Recognizing Change:
A Set of Short Studies in Pattern

by

James C. Forren

B.A. Studio Art
Wesleyan University, 1997

SUBMITTED TO THE DEPARTMENT OF ARCHITECTURE
IN PARTIAL FULFILLMENT
OF THE REQUIREMENTS FOR THE DEGREE OF

MASTER OF ARCHITECTURE
AT THE
MASSACHUSETTS INSTITUTE OF TECHNOLOGY

FEBRUARY 2005

Signature of Author: ___

Department of Architecture
January 14, 2005

Certified by: ___

J. Meejin Yoon
Assistant Professor of Architecture
Thesis Supervisor

Accepted by: ___

Bill Hubbard, Jr.
Chair, Department Committee on Graduate Students

Thesis Co-Supervisor:
Arindam Dutta
Associate Professor of the History of Architecture

Readers:
Mark Goulthorpe
Associate Professor of Architecture

Ann Pendleton-Jullian
Associate Professor of Architecture

Recognizing Change:
A Set of Short Studies in Pattern

by
James C. Forren

Submitted to the Depratment of Architecture
on January 14, 2005 in Partial Fulfillment of the
Requirements for the Degree of
Master in Architecture

ABSTRACT

This thesis presents a set of investigations on how pattern can be used to register change. Pattern - a set of attributes repeatedly transformed by rules - can be used to make apparent objects or ideas that transform and the operations that alter them. Our reactions to change, whether accepting or resisting it, are often repeated over time in a behavioral pattern. Repeated resistance to change is one of the patterns marking the psychological concept of Narcissism, a condition which can be both a useful defense mechanism and a paralyzing affliction. Faced with increasing technological change, contemporary culture has exhibited distinct patterns of resistance, patterns that describe a cultural condition of Narcissism.

This project examines the production of pattern through different mechanisms of change. Each mechanism distances the designer from decisions about the object's final form. This distance suspends a Narcissistic desire for control. Rather than controlling change, the designer's rules must adapt to it. This thesis does not use pattern as an imposed form. Instead, this use of pattern creates models for a reciprocal dialogue between the intelligence of the designer and larger processes of change.

Thesis Supervisor: J. Meejin Yoon
Title: Assistant Professor of Architecture

James C. Forren

Recognizing Change

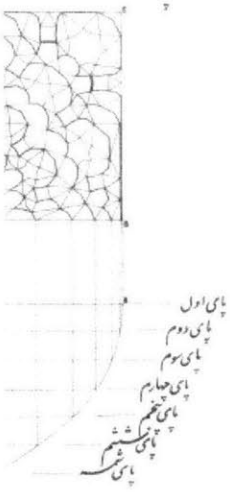
A Set of Short Studies in Pattern

Massachusetts Institute of Technology • Cambridge, Massachusetts

Contents

thank you,

Courtney Humphries, Susan Wadsworth, Christine Gaspar,
my Advisors, my Readers, Takehiko Nagakura


پای اول
پای دوم
پای سوم
پای چهارم
پای پنجم
پای ششم
پای هفتم

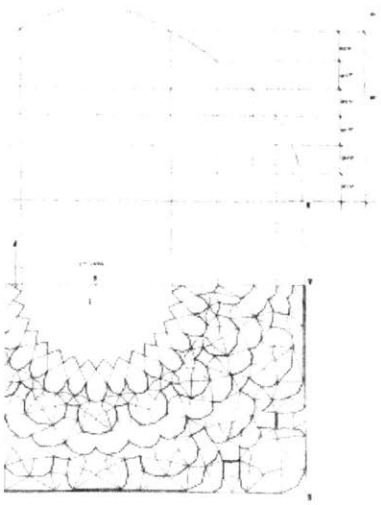
Introduction

The architect typically exercises considerable control over the final form of a design. This method of creation could be considered today as a narcissistic expression of creativity, one that denies change that is not directly designed by the architect. The exercises here represent a counterpoint to the narcissistic mind, generating patterns that play out in part without the direct manipulation of the designer. Pattern, a system of repeated rules and attributes, provides avenues for intervention. By incorporating external changes into its construction, pattern can become a vehicle for overcoming the narcissistic desire by the designer to control change.

I. Pattern and Architecture

Pattern can at once be an object in its own right and a template for making other objects. Architecture's use of pattern connects it to a broader family of visual culture and ideas. It is at once an aesthetic act and an applied art form or craft.

Pattern is used here as a generator for design. These are designs recorded primarily through line drawings. These drawings present the confluence of a particular design goal and the context which constrains this goal. For instance, an attempt at representing gradual change is restricted to a set of diagrams copied from a book.



Within the discipline of architecture design processes are engaged in the realizing spatial layouts and fabrication templates. By engaging a design process the architect applies space and material not only to the function of a shelter or a beam. The process of design introduces ideas into the formation of shelter. In this

way the ubiquitous objects of our existence – vessels, inhabitations – serve a second function as a rhetorical device. The meaning of an idea is embedded in the form of a room you sleep in or a wall you lean against. Through the perception of this wall or that room we also listen to the ideas of a designer and the culture they inhabit. As an object in its own right and a template for making other objects, pattern therefore offers insight onto a larger body of objects through the study of one.

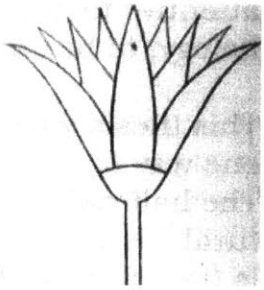
II. Pattern and Cultural Change

The "half-palmette motif" presents one such example. Alois Riegl, a late nineteenth century theorist, demonstrated in his work, *Stilfragen*, that this vegetal motif had evolved through different cultures (Schafter 47-8). Contrary to thought of the time, Riegl showed that ornamental motifs grew out of previous designs. It was thought that vegetal motifs were repeatedly copied from nature itself by different cultures, rather than passed between them.

One condition that emerges from Riegl's study is that the half-palmette motif could be altered to suit the different needs of different cultures. Riegl traces an exchange of the half-palmette motif between Greco-Roman and Islamic ornamental traditions. His history demonstrates the relationship between the ideas of a culture and the forms or inscriptions it generates.

Riegl traces a history of the half-palmette motif beginning with origins in Egyptian plant ornament. He identifies its pliability – the changes made and added to its form – as it is passed between numerous cultures. He cites Hellenistic and Roman cultures' half-palmette as nearing "direct imitation" of nature, a claim which is consistent with cultures pursuing extraordinary levels of natural stylization. Conversely Riegl discusses the half-palmette's adaptation in ara-

besque designs. Here the plant-form is stylized, adopting an "'unnatural' relationship between leaf and tendril" (Schafter 48). This stylization suited Islamic decorative traditions by eschewing direct representation of the leaf.



III. Cultural Change

Cultural change spawned the changes in the half-palmette motif design. Likewise, change in the half-palmette motif facilitated this cultural change. Like the half-palmette motif the discipline of design responds to and facilitates cultural change. It is directly involved in the inscription, transference and exchange of visual motifs like the half-palmette.

This inscription, transference and exchange is achieved through technological media. In this way the tools of technology play a key role in design, and through design, culture. Like the half-palmette motif's changes, the evolution of technology results from and facilitates cultural change.

This linkage from technology to design to culture is a reciprocal relationship. A change in one link affects the other two. Designers, then, are mediators between technology and culture. Designers embody the prosthetic relationship between the mind-body of an individual and their tool. The designer responds to and mediates the changes between one and the other, acting as the eyes and ears of culture and the texture and sensibility of technology.

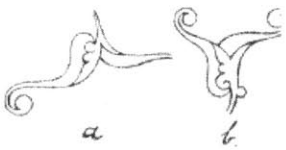
How then do designers see and hear for culture, and form the textures of technology? Recent studies in psychology and advancements in computation are signifiers of contemporary change in culture and technology respectively. Each has emerged as a recent development, or bud of growth, of these shoots. To exhibit awareness of these developments is to be

attentive to developing strands of culture and technology, to be abreast of their contemporary change.

This thesis suggests the use and study of pattern is one way to process and understand these changes. The half-palmette motif, both technological and cultural – is emblematic of the condition of designers. It is transformed through these changes and simultaneously facilitates these changes. Pattern, in a sense, is where designers enact their role, bringing technology and culture together. Pattern, also, through its rule-transformation structure offers an ideal form for bringing together contemporary technology and culture.

IV. Cultural and Technological Change

In *Understanding Media* Marshall McLuhan refers to the relationship between culture and technology in his chapter, "The Gadget Lover: Narcissus as Narcosis" (McLuhan 63). He recounts the story of Narcissus as emblematic of a culture's pathogenic response to technological change. McLuhan explanation of the Narcissus relationship is that culture suffers a fixation on the tools it has created. In a sense, Narcissus created the tool of the mirror in the surface of the pool. He then became fixated on the tool, mistaking it for himself.



McLuhan claims the same fate for cultures which do not – as Narcissus eventually did – reach into the pool and disturb its waters, cultures which do not extend themselves to find out what the tool they created is all about. What its capacities and characteristics are. McLuhan imagines such cultures as numbed creatures: somnambulists walking through the motions of its own existence.

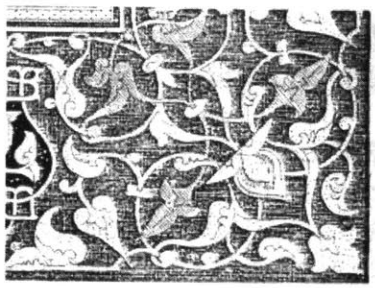
McLuhan's characterization of culture suggests that

individuals' responses to change can be loosely applied to culture's response to change. To understand McLuhan's argument more deeply we can consider Freud's development of the psychological idea of narcissism. McLuhan's description of technological change can be considered, in Freud's terms, as "excitations." In the behavior of individuals, there are two chief responses to change, or excitation. One is to account for it or adapt:

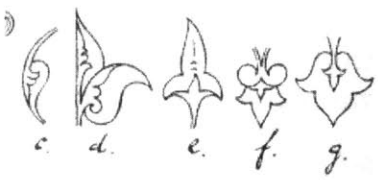
"We have recognized our mental apparatus as being first and foremost a device designed for mastering excitations which would otherwise be felt as distressing or would have pathogenic effects. Working them over in the mind helps remarkably towards an internal draining away of excitations which are incapable of direct discharge outwards." -Sigmund Freud, *On Narcissism: An Introduction*, p.85

The other reaction an individual can have is to ignore or resist change, to turn "away from the external world – ...from people and things" (Freud 74). This response, according to Freud, represents pathogenic behavior.

By reaching into the water, disturbing and discovering its nature, Narcissus came to realize the image he saw was actually a reflection of him. This realization did not ultimately save the youth. However, Narcissus did come to understand that it was the *image* of him which captured his attention, not the vision of someone else (Ovid ln 672-674). Despite his ultimate failure to free himself from this fixation, by disturbing the waters Narcissus did engage a mental processing. In this way, the physical play with and exploration of a medium develops to mental processing.



Therefore, the mental processing of cultural change requires play with the media of this change, technology. Contemporary technological change is characterized by computation. To play with computation is to develop frameworks for characterizing and pro-



cessing cultural change. Computation is a tool processing vast quantities of information in seconds. Mental tasks which once took days, such as complex and numerous calculations, now take seconds. In a sense, in relation to pattern, vast quantities of attributes and their corresponding rules can be synthesized at before unknown rates. The result is that objects described above – patterns, inscriptions – objects of self-representation are immaterial. The form itself is transient. What endures is the attribute-rule relationship.

This vacillating mode of self-identification corresponds to psychological negotiation in the face of constant change. Just as forms on a screen can number the thousands in minutes, the individual can cycle through multiple identities in a brief amount of time. This iterative identification becomes indicative, then, of contemporary cultural change.

Pattern's structure of rules and transformations can be a way to mentally process the characteristics and transformations of cultural change. It is a way, like Narcissus reaching into the pool, to engage this mental process through a physical medium. Riegl expressed such a relationship between physical material and psychological need:

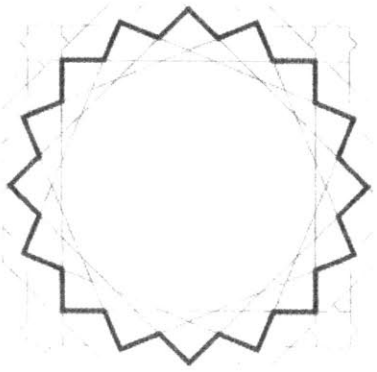
"Ornamental development in Riegl's theory... is the result of a material organizing process that transforms existing motifs over time in order to serve new psychological needs.." (Schafter 47).

Pattern, an art-form that (like the half-palmette motif), changes over time, acquires its pliability through attribute-transformation relationships. Multiple forms can take advantage of and be derived from this relationship. It is, therefore, an art form (or non-form, but, instead, rule-set) uniquely suited to change. It is an art form that can uniquely grasp and manage the need for negotiated definitions.

This thesis proposes that to serve this end today, single patterns must change and negotiate their form. Traditional methods emphasize a single permutation of form repeated throughout the pattern. But psychological behavior is understood only through iterations of varied, related permutations. Therefore, to be capable of engaging a productive mental process, the physicality of pattern must deal with iterative, but related forms. The single form, in psychological and computational processes, is not enough to achieve understanding.

Additionally, ideas external to the will of the designer must participate in the negotiation. As stated by Freud, narcissism represents a turning away from the external world and its objects. Therefore, mentally processing excitations requires an external object. This is a force external to the mental process. It is a seed around which to shape the mental organizations.

These studies introduce external forces, drawing both on the inheritance of pattern itself (as suggested by Riegl) and on the simulation of natural forces. This latter method takes its cue from another late-nineteenth century theorist, John Ruskin. For Ruskin the ruination of ornamental form by natural phenomena, as well as the derivation of this form from natural models, symbolized the highest order of divine law (Schafter 18-19). Like Riegl's theories of historic form (the trace of historic consciousness), Ruskin's argument required the intervention of a larger, divine consciousness. For ennobling and evolving culture, both arguments necessitate the intervention of external form.



V. Change and Pattern

What are the ways patterns can describe change and introduce external forces into its development? First, we must look at methods of distinction and transfor-

mation in traditional pattern-making. Within the vocabulary of Moroccan Arabesque pattern designs are a series of different ways to distinguish one pattern from the next. A common family of pattern designs starts with the initial shape of the square (Castéra 100). To distinguish different subsets of this family the square is rotated a different number of times. A square can be copied and rotated 2, 4, 6, 8 and 16 times to define different classes of patterns. The subsets are distinguished through the number of transformations they undergo.

The model of Moroccan Arabesques distinguishes one pattern from the next. However, to generate change within a single pattern the attributes (the square) and transformations (rotation) must vary inside the single pattern. First there must be some kind of stable form; an attribute or set of attributes which does not change. For instance, this can be a grid whose dimensions remain constant, or a shape whose basic organization remains the same. Second is the form that changes. An attribute must undergo some sort of transformation within the domain of the stable form. For instance, a shape within the space of the grid can change from one shape to another, or the dimensions of a shape can change. The eye then reads the changing shape against the unchanging context.

We can use these rules to describe change. For instance, the very rules that determine the pattern can describe a change occurring in the shape. A shape can become larger or smaller in response to its coordinates within a regulating grid. A pattern can also change by representing natural phenomena. For instance, by isolating an attribute of light-change like the quantity of lumens collected by a surface, we can assign a perceptual signifier – a line, a shape, a mark – to this attribute. In making this assignment we can then alter the attribute according to laws of its construction: a line can become longer, shorter, thicker, thinner, curved, straight, etc. in response to

the number of lumens measured.

VI. Change and Design

The pages here will outline 5 designs. Each design distances the designer from the process in a different way. Some relinquish control over the initial form; others turn over the middle or ending of the process to external controls.

The goal of this investigation is to explore ways of engaging the designer in a fluid exchange between external rules and internal desires. This is to create a framework for a reciprocal dialogue between the design and the designer. The efforts here can be thought of as interventions, through rules, in a process that engenders growth.

No single study entirely accomplishes this end. However, each study points to a process, rules or both that contribute to this goal. These may be an environmental phenomenon like light or a set of rules established and followed by the pattern itself. These opportunities occur at various stages in the design. The future goal of this investigation is the incorporation of these opportunities into a single study with multiple avenues for external intervention.

This instruction in visual methods and knowledge is crucial to the foundation and development of creativity. Machines and processes may inspire and expand our ideas, may help pass these ideas along, but they cannot originate them. Nor can our ideas find form without the machines. This prosthetic exchange, the dialogue between internal desire and external constraint, is, therefore, the subject of this thesis.

Pattern Studies

1 About the Pattern Studies

The goal of these studies was to integrate some element of external control into the design process. This occurred progressively over the course of five studies; from manually copying and combining given forms to establishing a set of rules achieving a self-defined form. The set of studies uncovered several frameworks, or stages, for introducing external controls in the design process. However, no single study managed to tie these stages together.

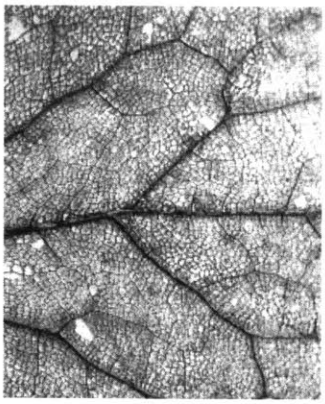
Each pattern study began with a specific goal in mind. Each goal pointed to a specific method or form from which to derive the first steps of the study. These goals ranged from responding to the change in light exposure on a surface to responding only to rules of pattern-making itself. In this way the investigation progressed from literal design solutions to more abstract studies of pattern itself.

Pattern-making consists of two primary components: the unit and the pattern itself. The pattern is formed through combining units. Unit definitions and changes affect the overall pattern. Likewise, transformations to the overall pattern can augment the units. Some of these exercises examine transformations applied to units. Some examine these changes made to the overall organization of the pattern.

Transformations play a large role in these studies. They are a primary way to represent and instigate change within a pattern. These transformations are recognized as a deviation from the overall pattern and provide complexity to its organization. Traditionally they occur as counter rhythms to a primary one (Wolfram 42). As well, like the pattern itself, they are typically defined by the designer. In addition to units and patterns defined outside the designer's

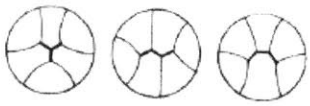
control, this thesis seeks to achieve externally defined transformations. The transformations studied here are location, relocation, rotation and mirroring.

Finally, the direct role of the technology of computation in this investigation is important to consider. The computer has been used in each investigation, from a routine copying and arraying device to carrying out sets of scripted rules. The final studies rely on computation to achieve their final form.



The frameworks of unit and pattern; attribute and transformation are instrumental towards integrating pattern with computational processes. The scalar distinction of unit and pattern provides points for describing the rules of a form without giving it final definition. The relationship between attribute and transformation provides a syntax for describing these rules. These frameworks open a door towards combining pieces of each study to developing further, more elegant designs.

2 *Hexagon*



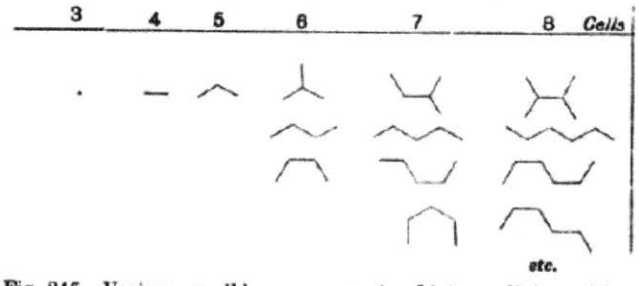
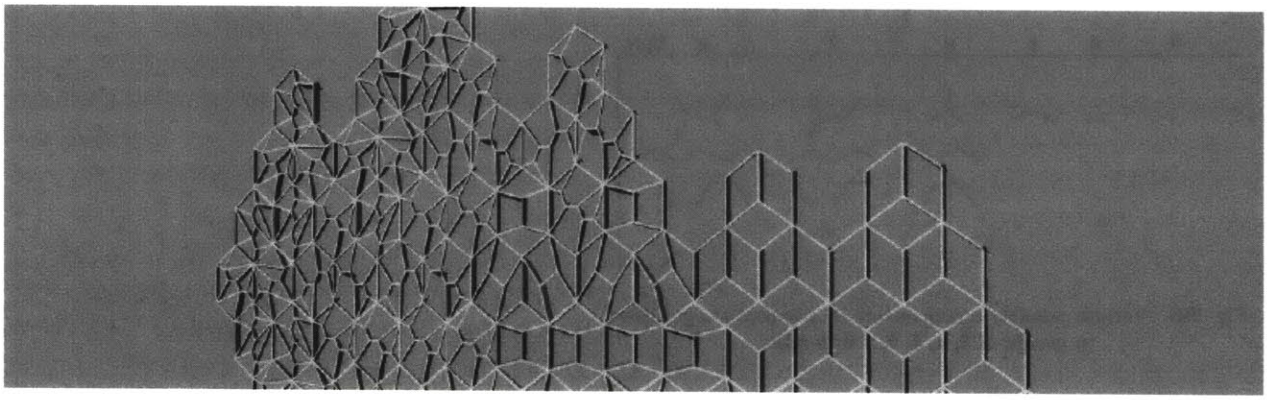
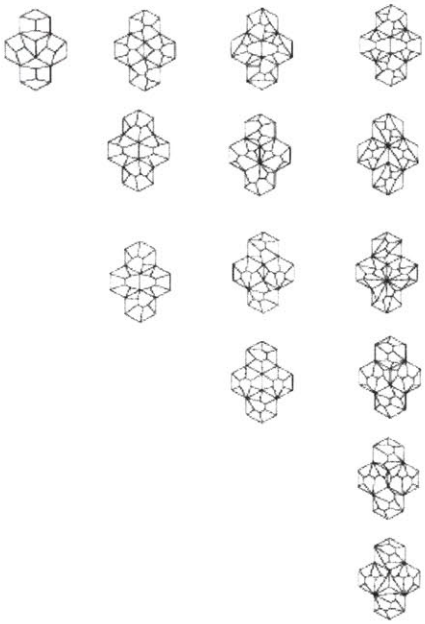


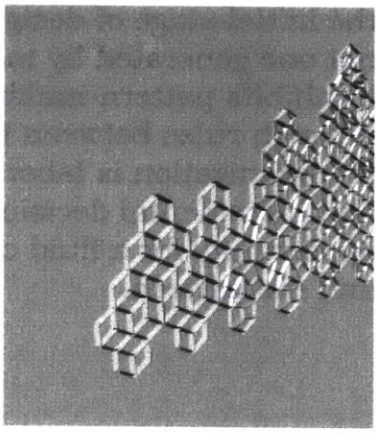
Fig. 245. Various possible arrangements of intermediate positions, in groups of 3, 4, 5, 6, 7 or 8 cells.

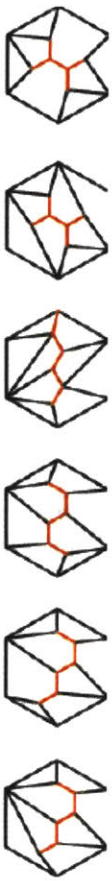
The hexagon with interior lines was chosen as an initial model of gradual change. This is because it is an organic model. It represents a starting point; an origin of gradual change.

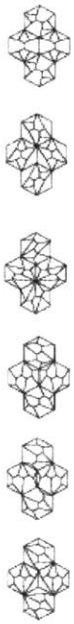


The goal of this study was to release the designer's control over the initial unit form. This was achieved by copying a form not made by the designer. The layout of the pattern was controlled by the designer. Despite this exercise of larger control the intention was to achieve an unexpected form through repeating the unit. The pattern, thus, obscures each individual unit. The conjunction of units makes many different sub-forms, until the denser areas lose all description of distinct units.

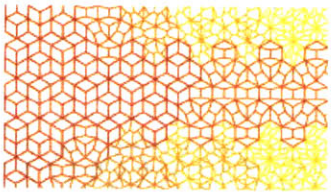


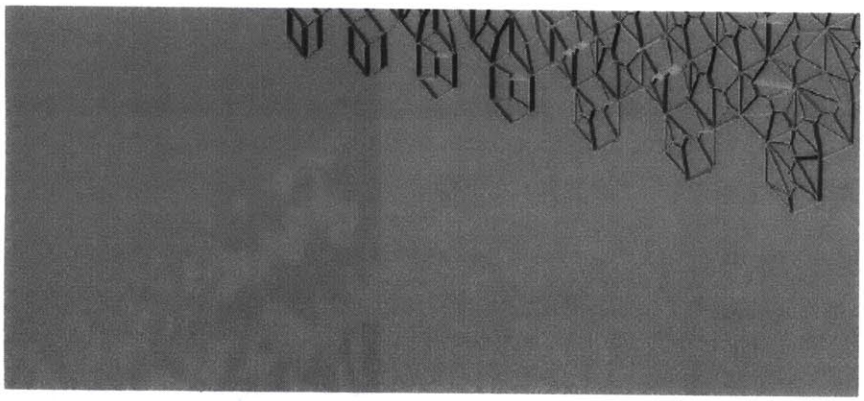




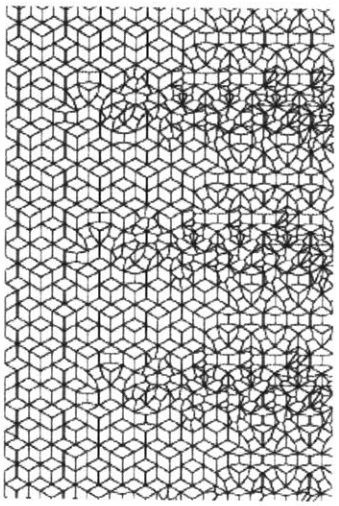


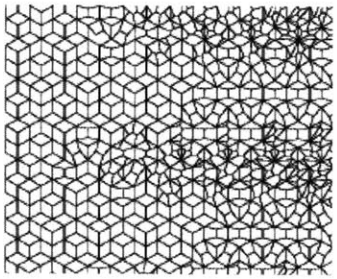
Within the stated goals of the thesis, the release of designer control is very isolated. It occurs at only the initial stage of design. It is also a copied form, not one generated by independent rules. Therefore it inhibits pattern-making unit inhibits a dialogue through rules between the designer and the pattern. Its organization is laborious and requires repeated and incremental decision-making. There is little opportunity for a fluid exchange between the designer and the pattern.

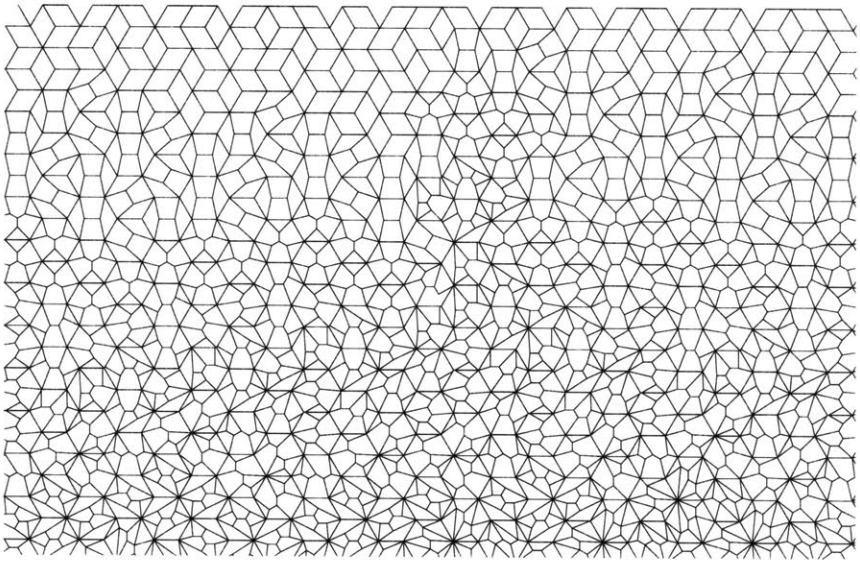


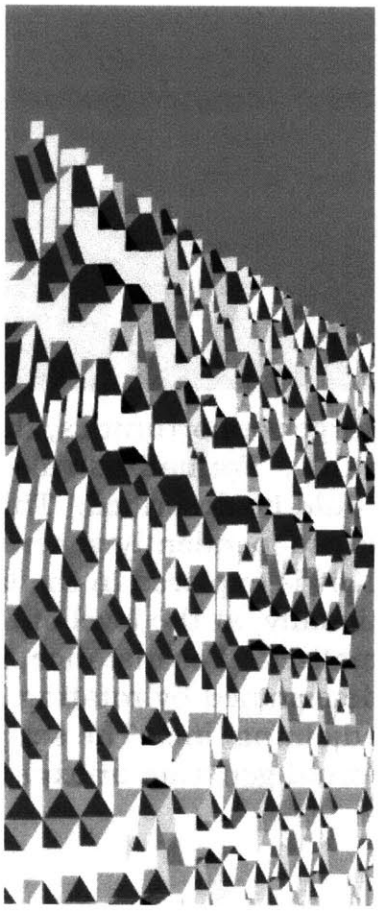


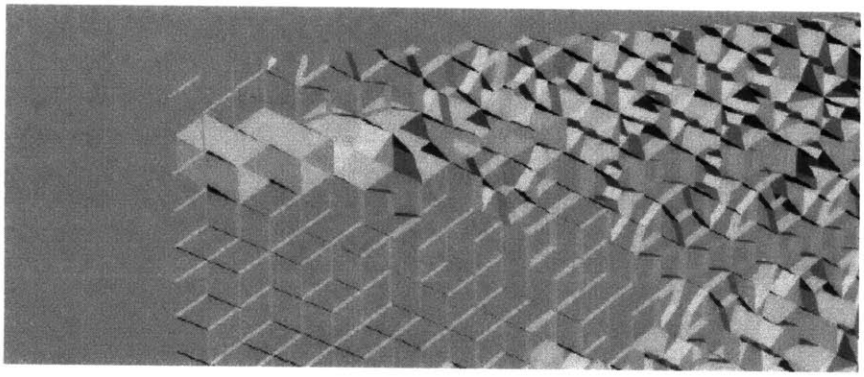
Larger opportunities, then, are suggested by the study. Formalized rules about the larger pattern could be described. This would allow a computer to independently generate a series of various forms. This is explored in the later exercise, *Triangle*. Additionally, at the level of the unit rules could be formalized. For instance, the position of each line is determined by rotating it by 120 degrees from an adjoining line. The scripting of this simple rule alone would generate unexpected forms.

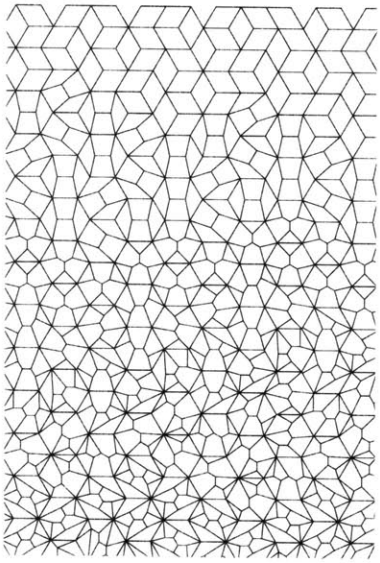


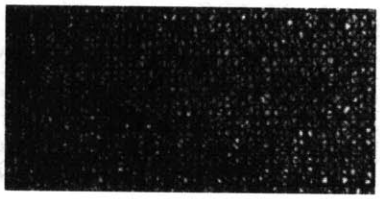




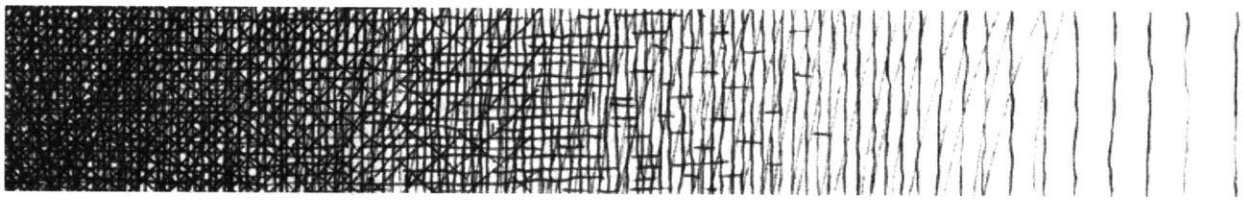




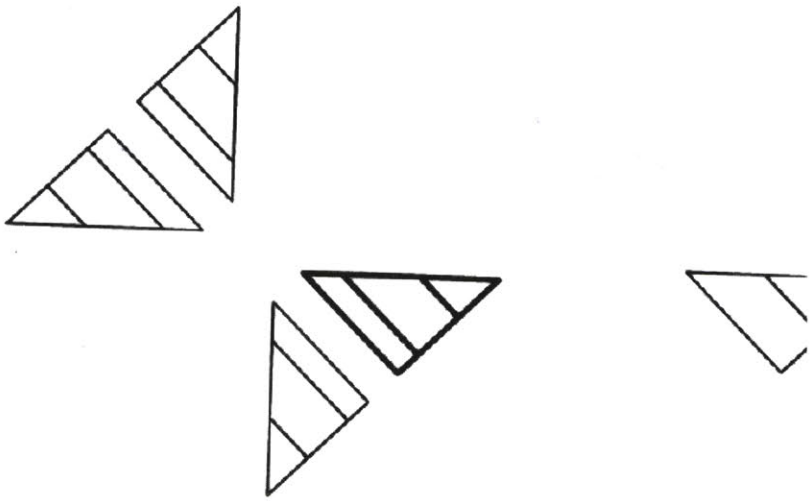




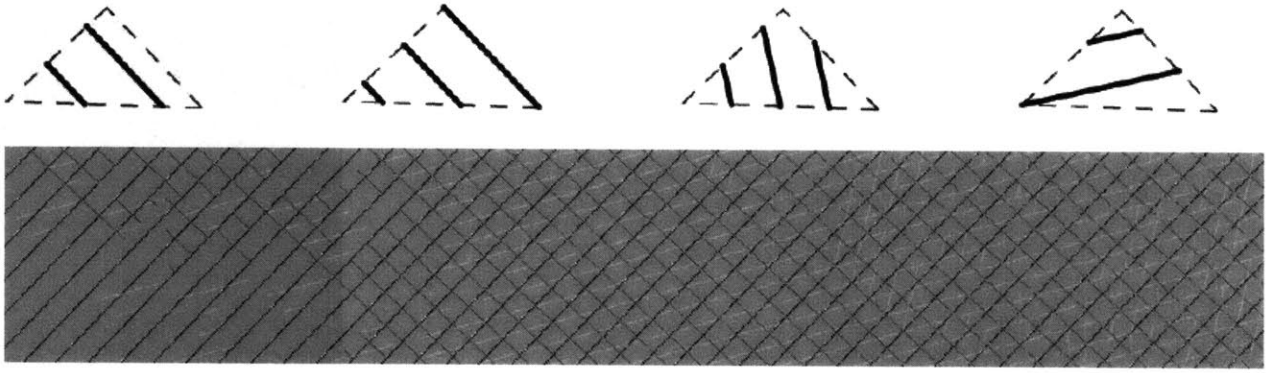
The subsequent investigations would look at trying to find rules independent from a pre-imagined geometry, rules which could be manipulated to achieve variable, but related, forms. Thus, rather than copying a diagram the next two investigations would attempt to derive independent form from the study of natural phenomenon. Natural light was selected, inspired by the value of light in these three-dimensional renderings.

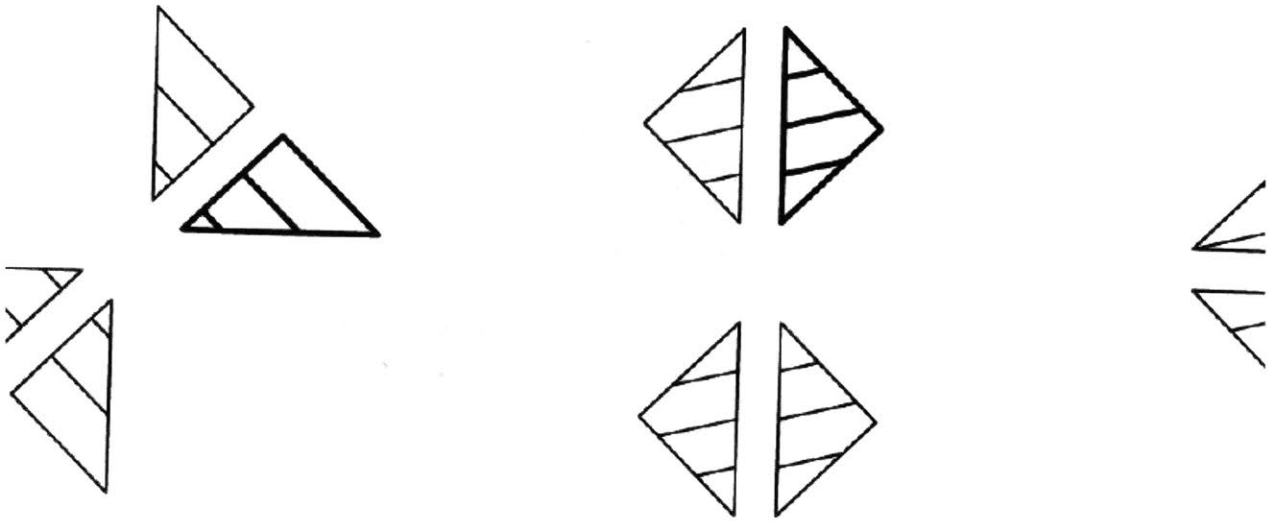


3 *Hatch*



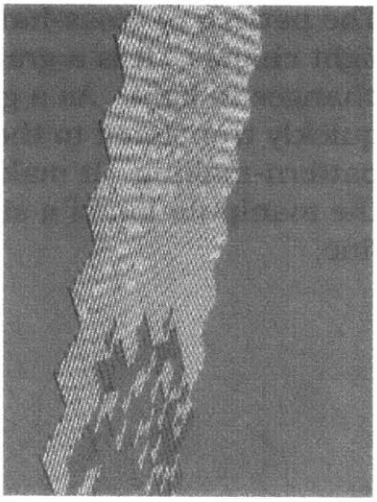
The pattern of cross-hatching is a model of gradual light change. It is a graphic representation of changes in light. As a graphic description it can be quickly translated to the two-dimensional activity of pattern-making. It makes this translation through the manipulation of a simple, singular element: the line.

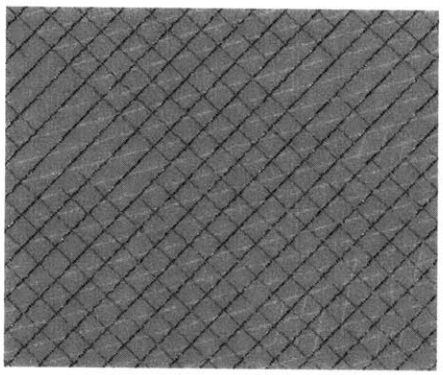


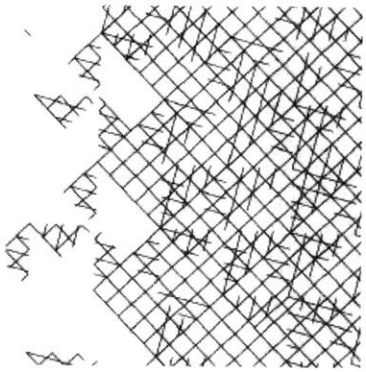


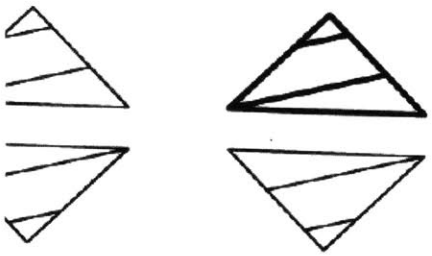
The goal of using hatching was to establish rules for unit-making. Rather than copy a form without establishing rules, this study established initial rules which would determine the unit's form. These rules, then, provided unpredictable forms.

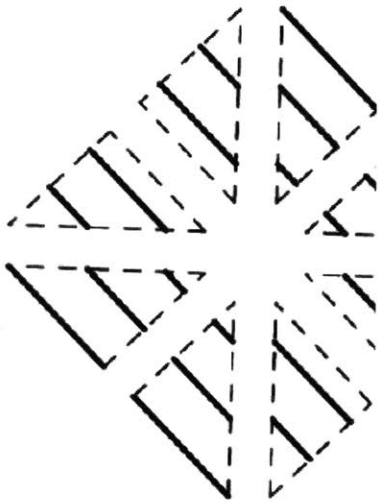
The imposed form for the unit was that of a rotated



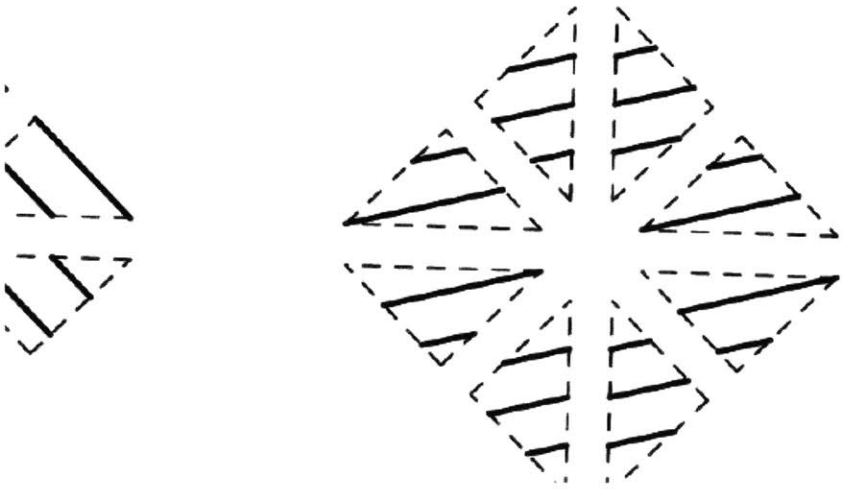


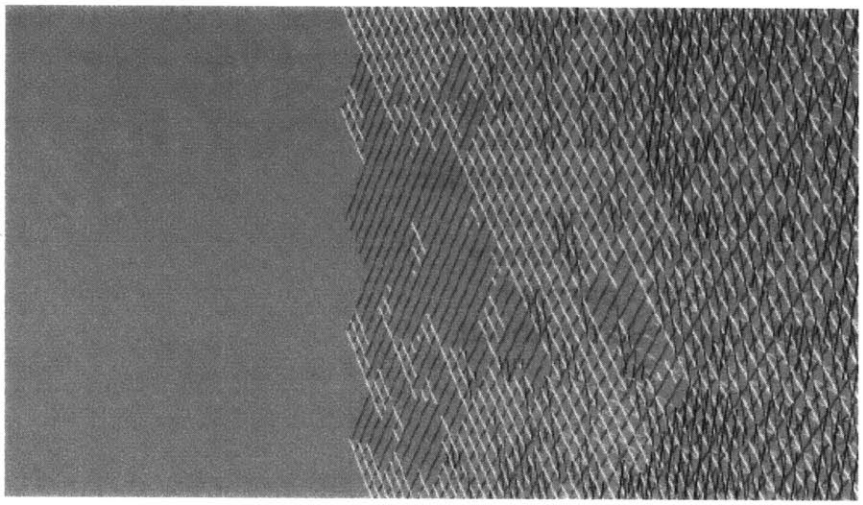


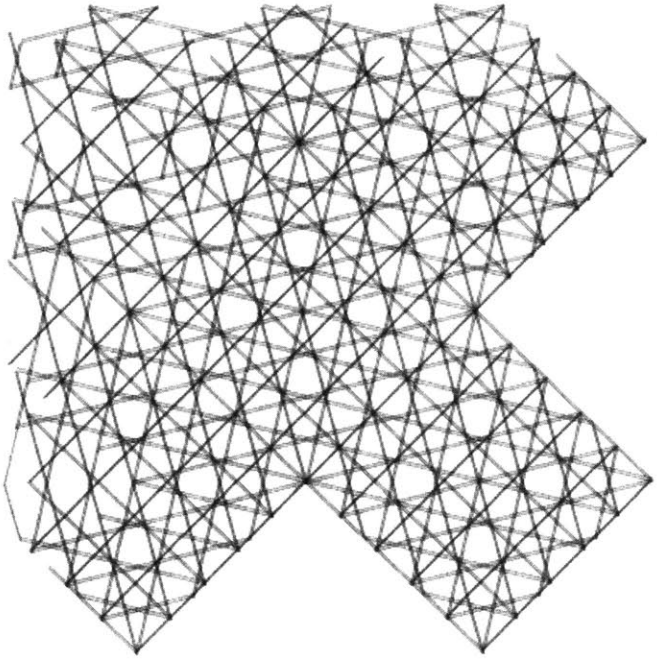


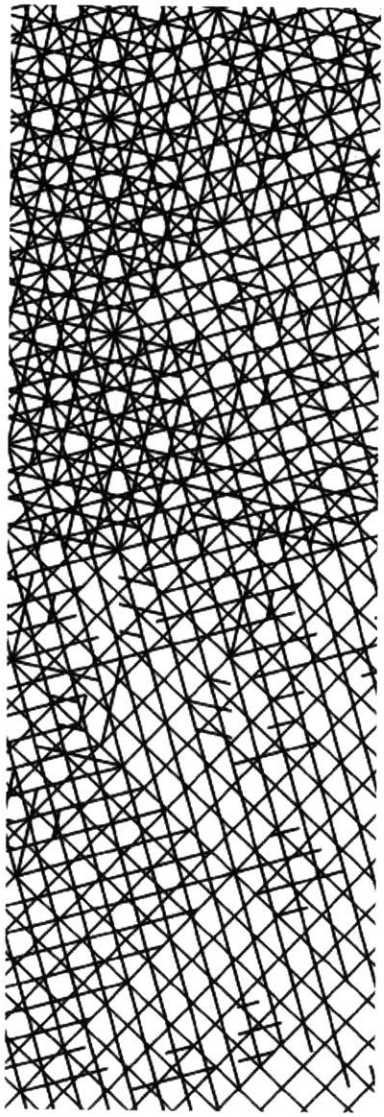


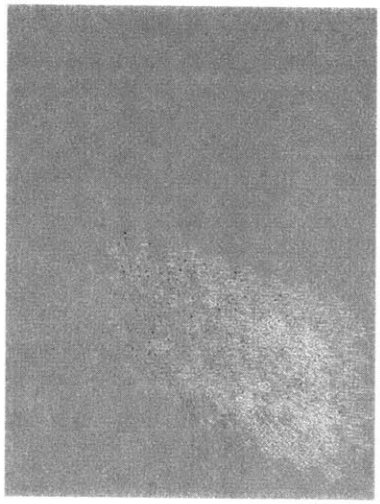
grid. The grid orientation, however, was derived from a particular light angle of daylight. This is a first rule of the pattern. The unit shape was imposed by the designer. It was selected from the units of the grid. These units were then broken down into smaller triangles in order to achieve a gradual progression from dark to light. Finally, there were two primary unit rules. First, all lines in the unit are parallel. Second, they flow from one unit to the next without interruption.

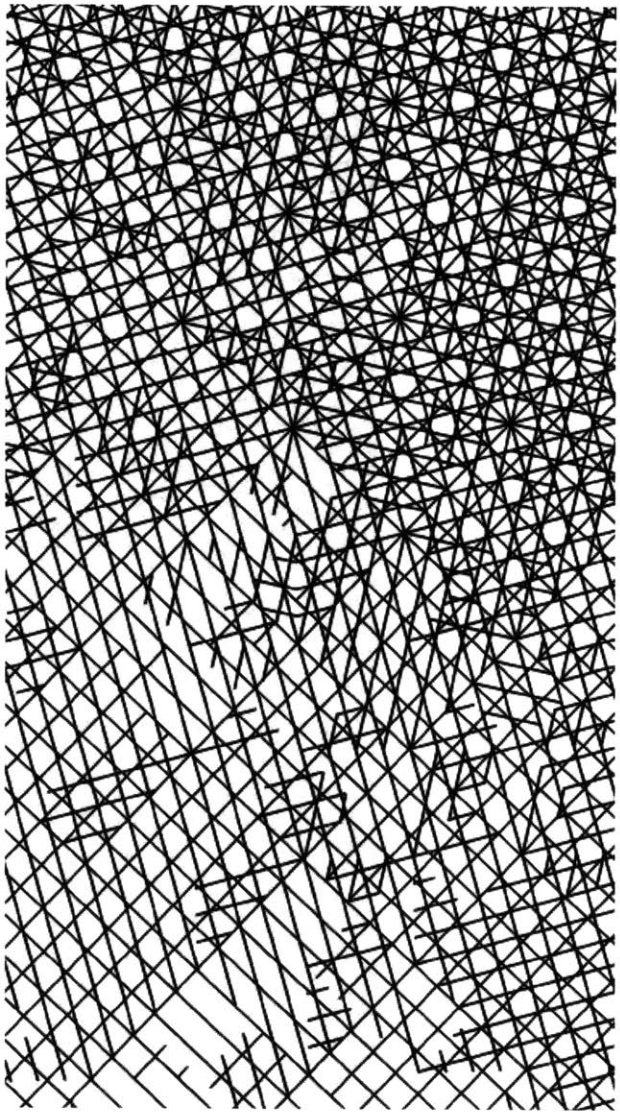


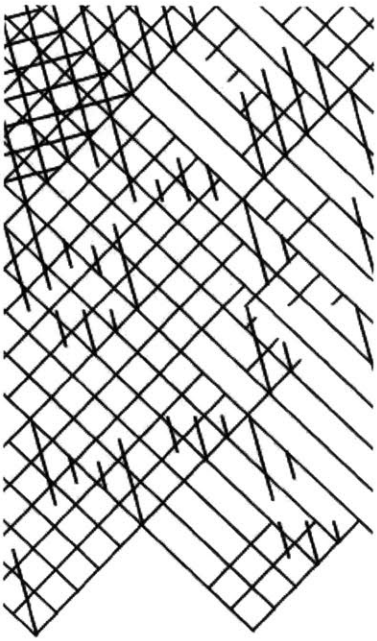


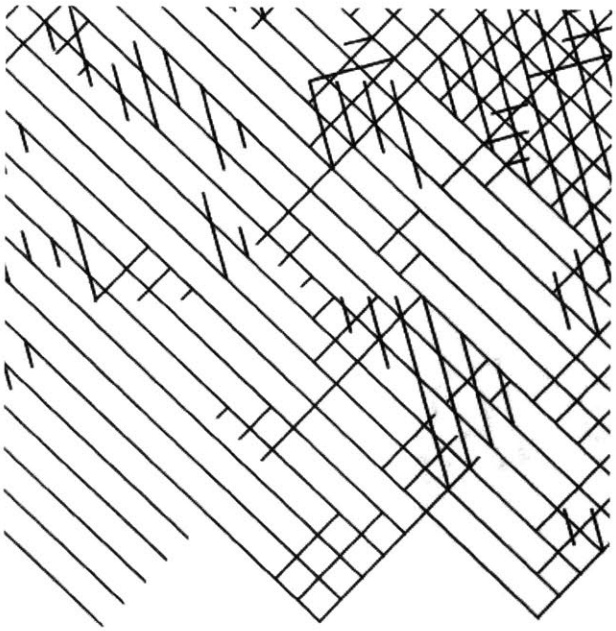




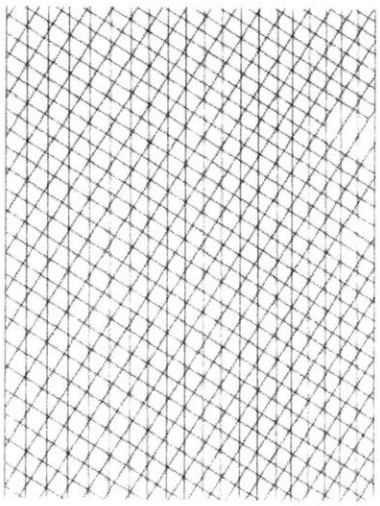


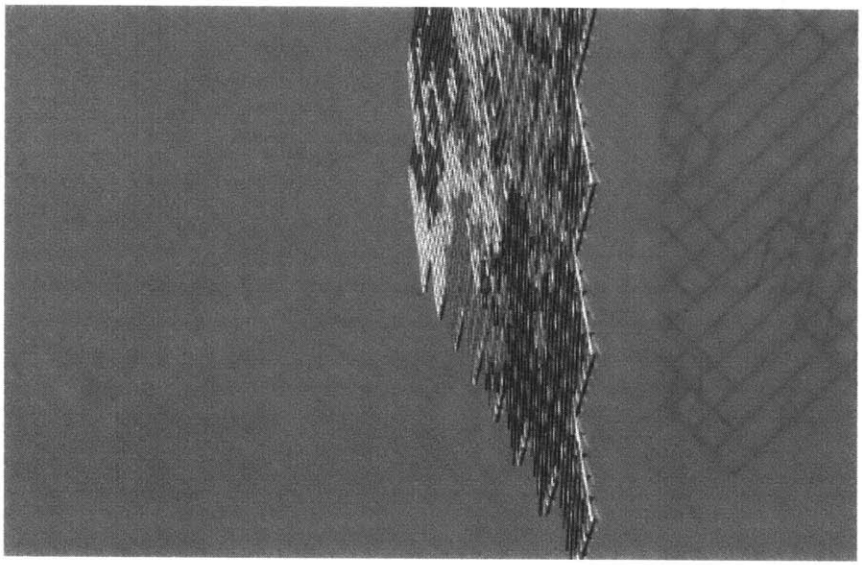


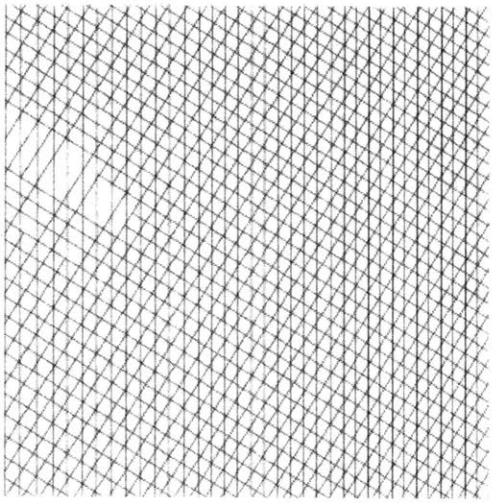


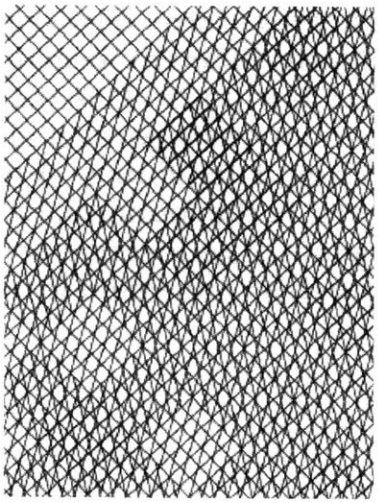


These patterns have computational implications. Firstly, different formalized rules can be applied to the four units. One could be relax the restriction on continuously parallel alignments, generating snaking triangulated forms. A fundamental exploration, however, is the foundational rule itself: the continuous points of connection between four shapes, maintaining this connection through full rotation of the units.

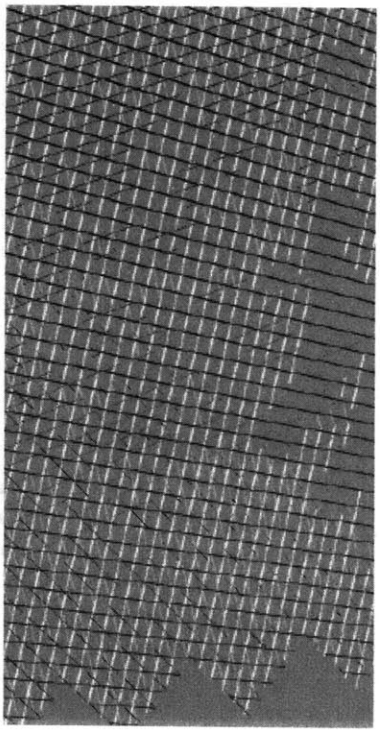


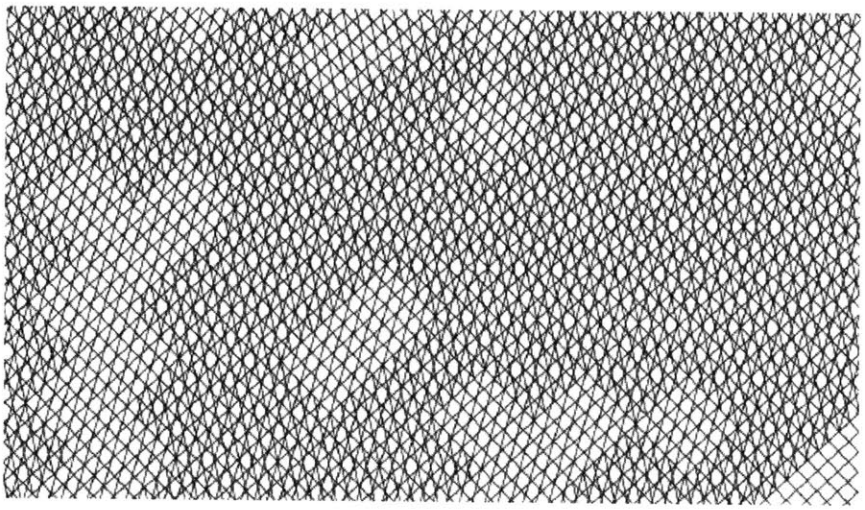


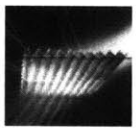


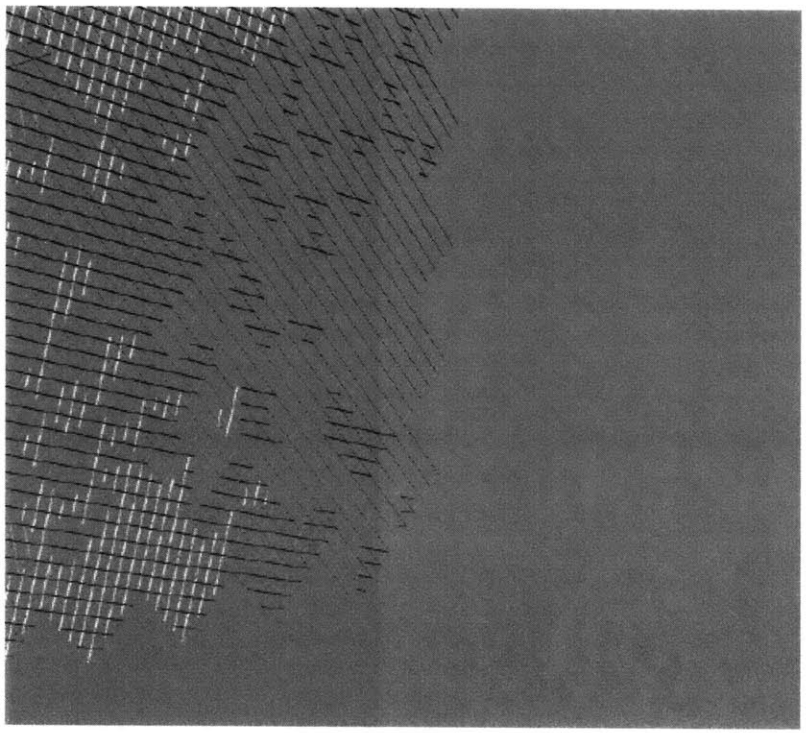


The rules of light filtration in this study primarily served the graphic form of hatching. The next study continues looking at the external condition of light, but using analytical, not representational rules. These rules are suggested by the phenomenon itself, not the representation of this phenomenon.

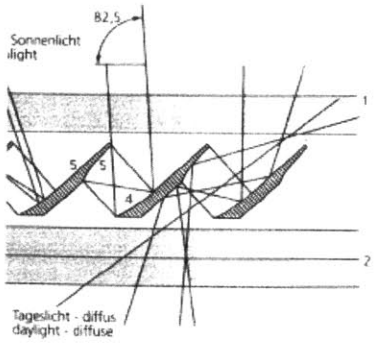


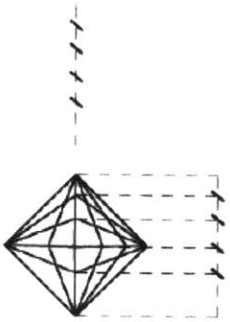


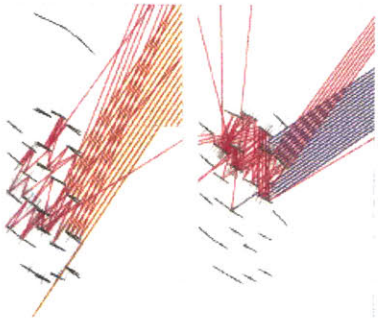




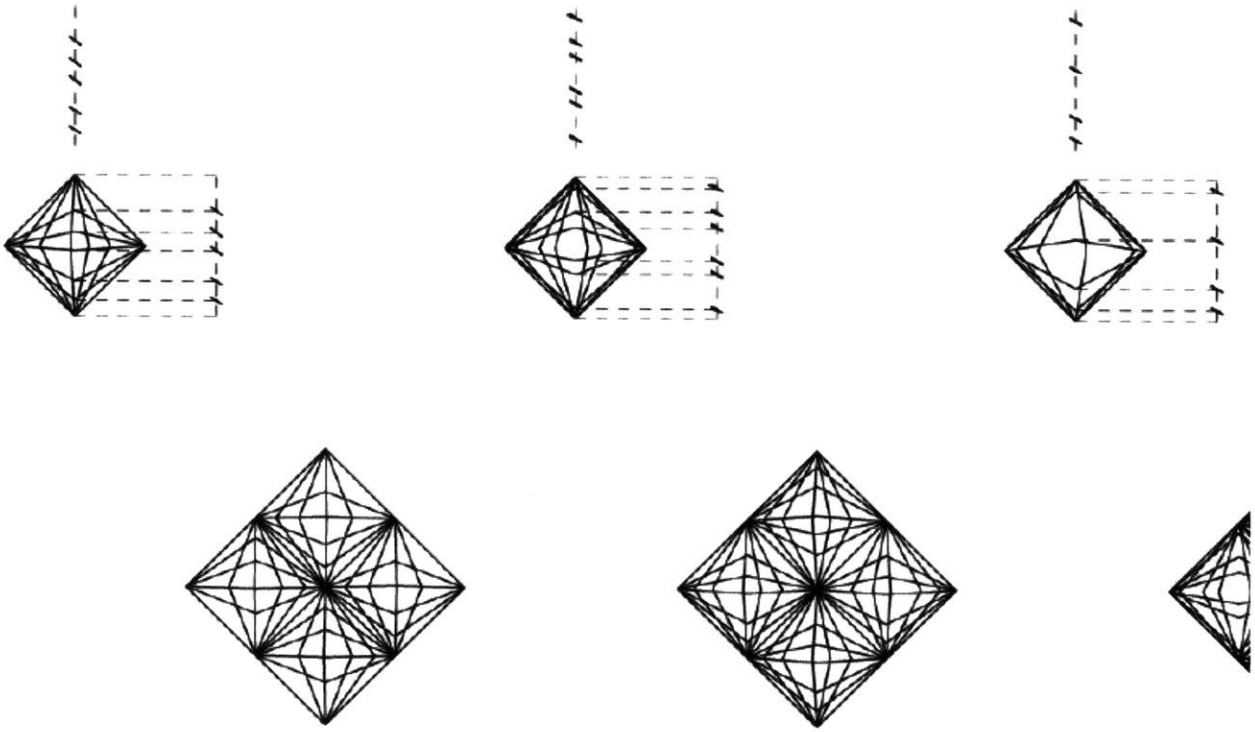
4 *Diamond*



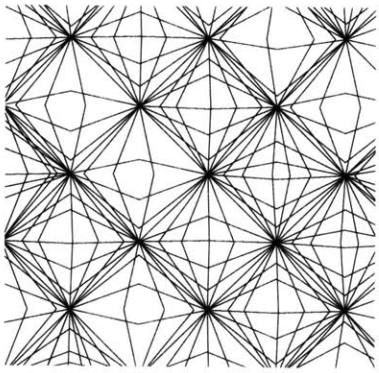


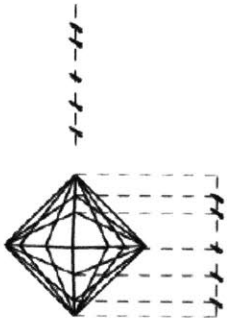


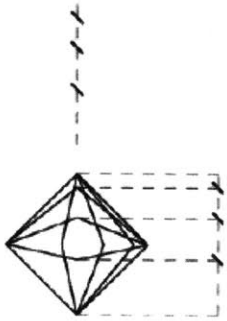
The diamond with interior lines is the first three-dimensional pattern. The previous patterns started as two dimensional drawings and were then given three-dimensional form. This study begins by looking at the final three-dimensional form.

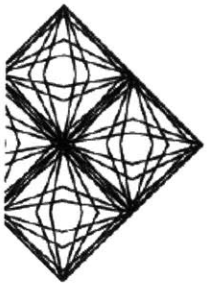


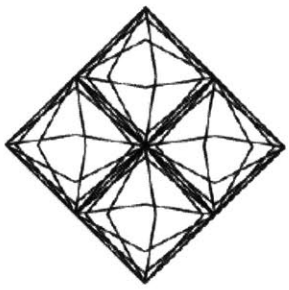
This pattern is a result of light analysis. It uses information about light's behavior to establish rules about the pattern's form. Light in this exercise is treated as a vector: a line guided in one direction. This is a simple abstraction of light's propagation. Light's propagation is studied as it passes through several vertical planes. Each plane is described by a set of "fins": lines representing surfaces that redirect the rays of light.

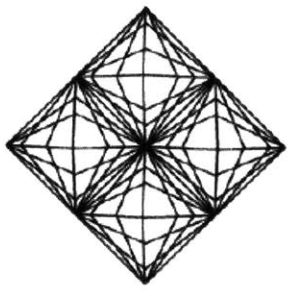




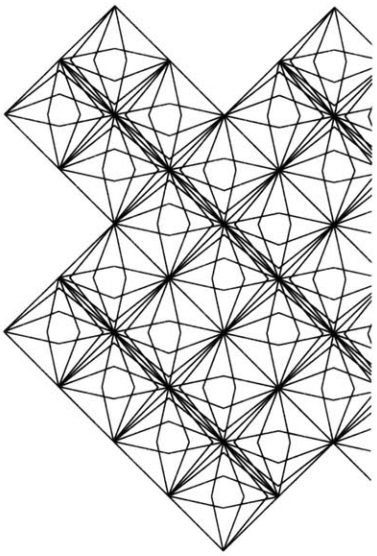


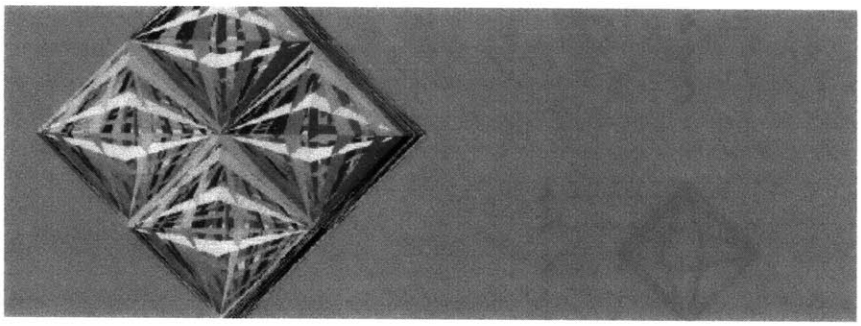


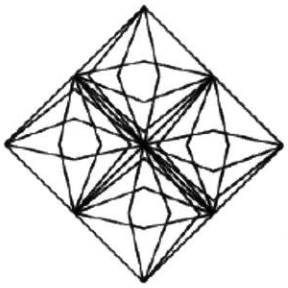




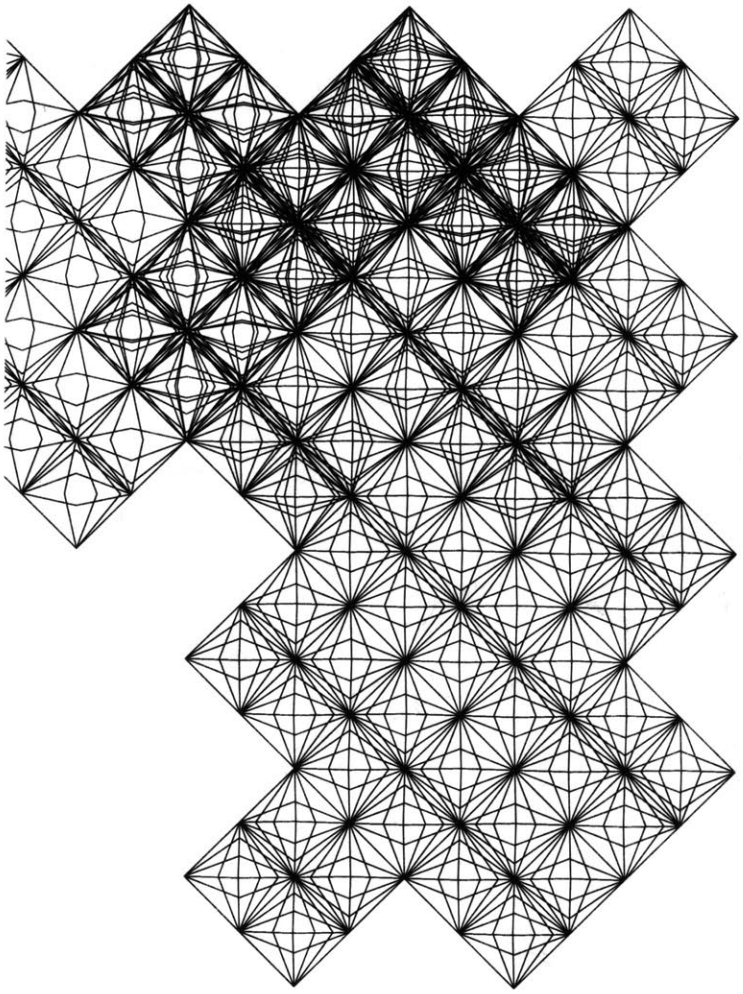
The control established by the designer was to admit as much light as possible through these surfaces within a specific angle range. At the same time, these fins were to reflect as much light as possible within a higher angle range. The fins' were located according to a) how close they were to adjoining fins (normal from the reflective surface) and b) the best position to block high incoming rays.

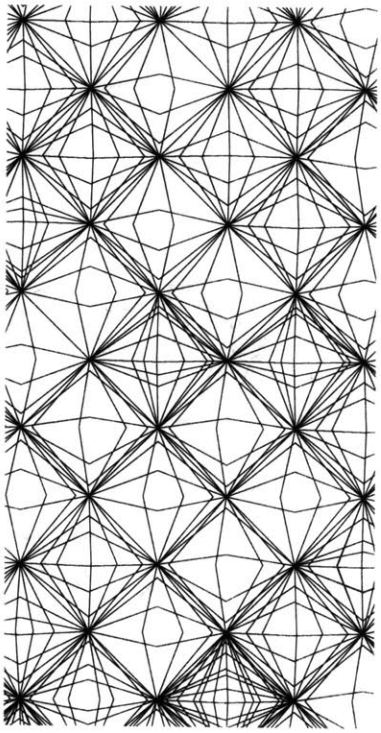




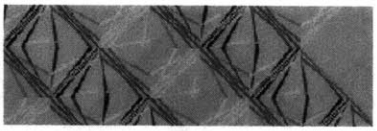


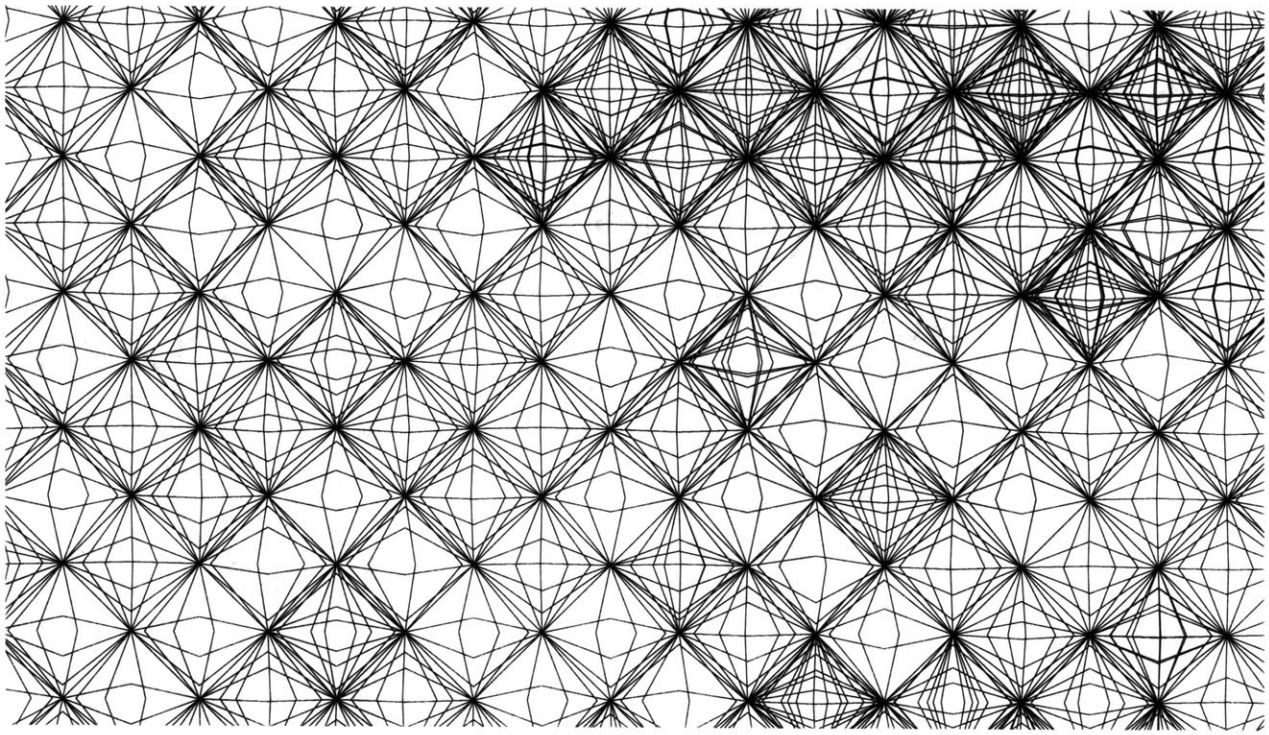
The form of a diamond was then imposed on these sections. The positions of the fins were located along the diamond's central axis. Lines were then drawn from these points to opposing corners of the diamond. This form was then copied and rotated to make the final unit, establishing a cross-grain.



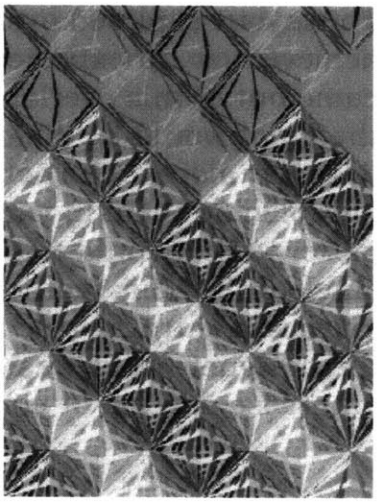


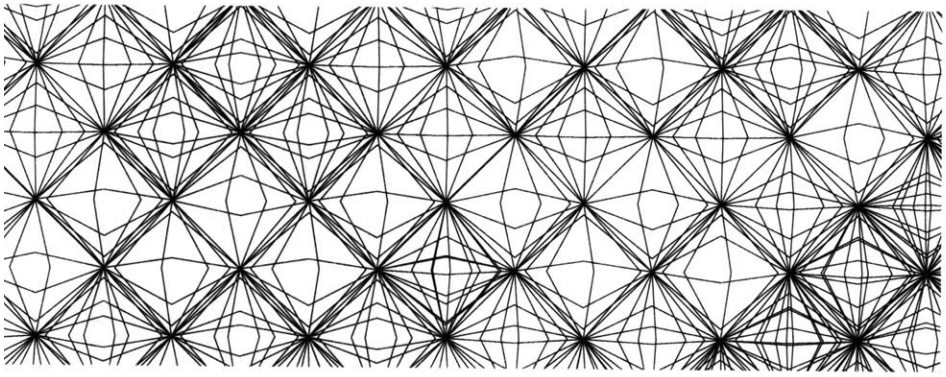
The layout explored overlapping different diamonds to achieve new, unexpected units. Unexpected shapes came out of this overlapping. The fineness of the grain would be difficult to achieve through the designer alone. The act of defining separate planes and placing them in various combinations created patterns that could not have been described in a single unit. This method of overlapping is a useful external control: a rule that, although imposed by the designer, yields results the designer could not otherwise develop or control.

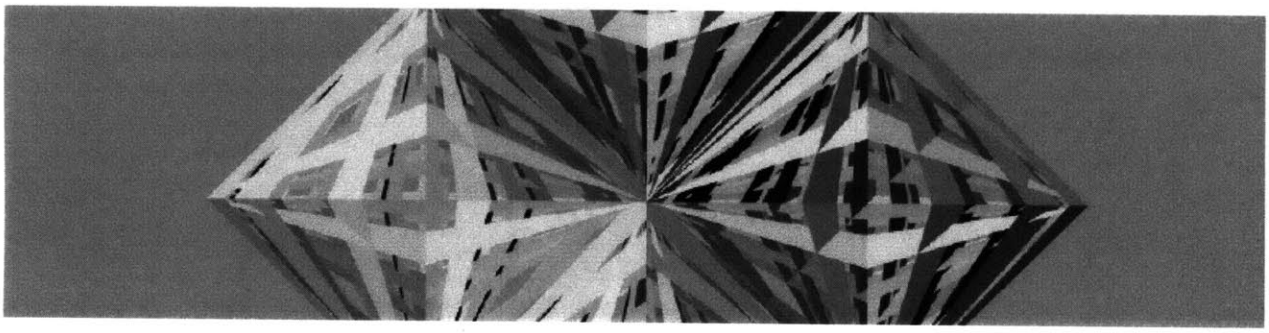




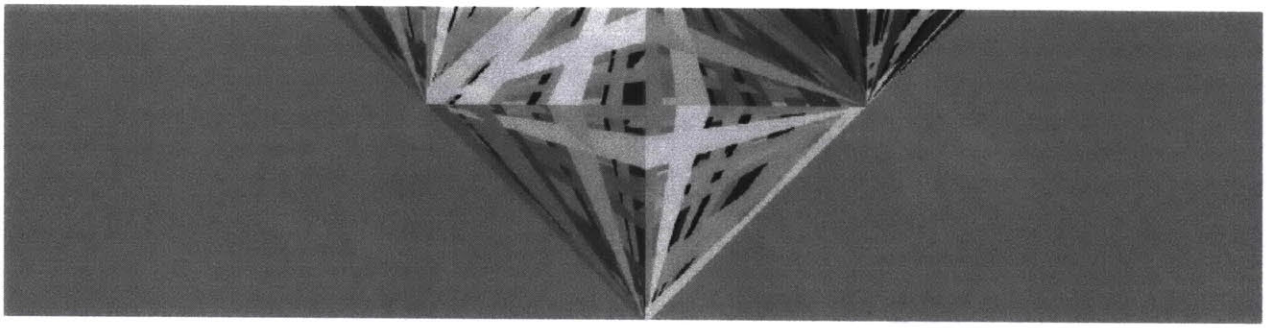
Several rules beg for further exploration. First is the generation of each individual unit; i.e. the placement of points along the central axis. These positions can be determined through a number of avenues. The light study is just one of many games one could play with their position. Additionally, rules about the shape of the imposed form – the diamond – require further study. Finally, rules for overlapping the diamond units can play a primary role in further development of the pattern. The arrangement of the pattern was based only on incremental decisions by the designer.



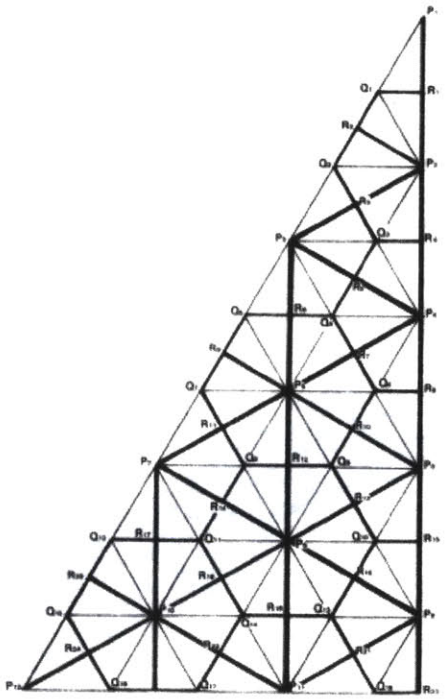




This study represents the first clear set of opportunities for rule-generation and intervention at many levels in the design process. The next set of exercises starts to look at the idea of changing patterns based on the interplay between the rules of layout and the rules of unit formation.

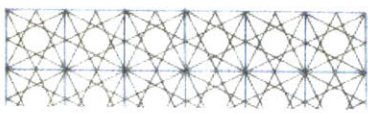


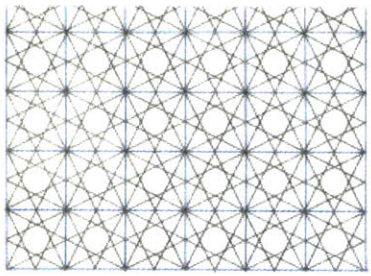
5 *Triangle*

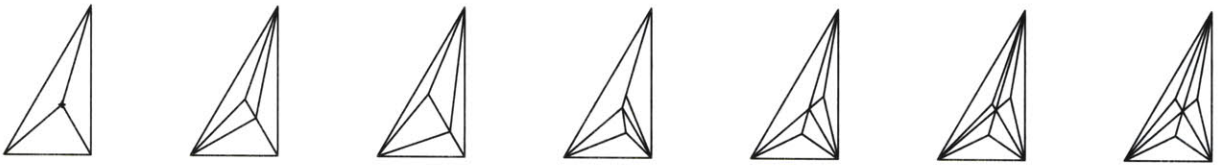


In order to explore the interplay between pattern and unit, these studies draw from established traditions of pattern-making. The triangular unit, -- extrapolated from the grid of Moroccan Arabesque patterns -- was combined with the Cartesian geometry calculations of AutoCAD software. The triangle suited a Cartesian coordinate system and trigonometric calculations and facilitated locating the pattern in three-dimensional space. This was a step away from the previous explorations' isolation to a two dimensional surface. The rules for cell-division used in the *Hexagon* exercise -- a familiar model in developing gradual gradation -- were adapted here to the geometry of a triangle.

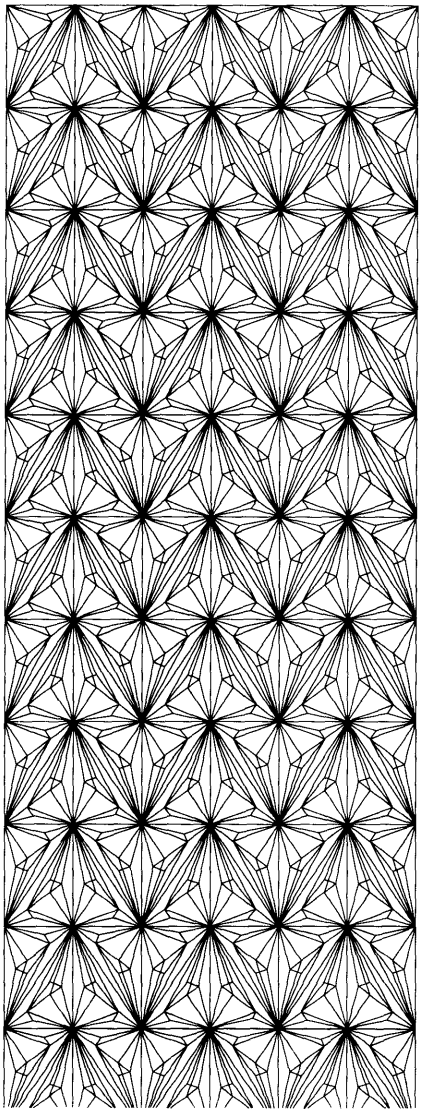
Triangle is the first set of patterns described through code, not manual drawings. The rules were formalized into a set of commands. They were defined so that any 3 points in space would yield some permutation of the triangle form. This allowed for rapid feedback between the designer and the pattern rules.

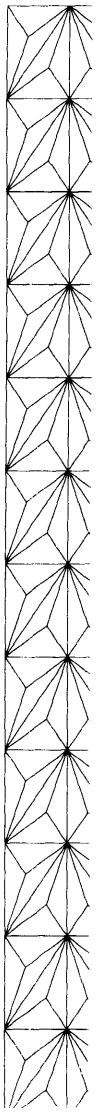


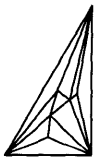




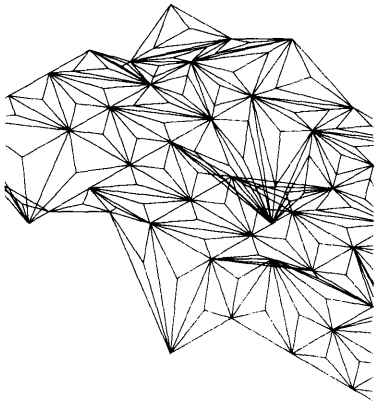
Describing the forms through rules released the designer's control over the final form. Parameters could be adjusted, but the designer was totally unaware of what the results would be until the pattern was completed. Over time, familiarity with the rules allowed for a tighter dialogue between the designer's expectations and the tool's descriptions.

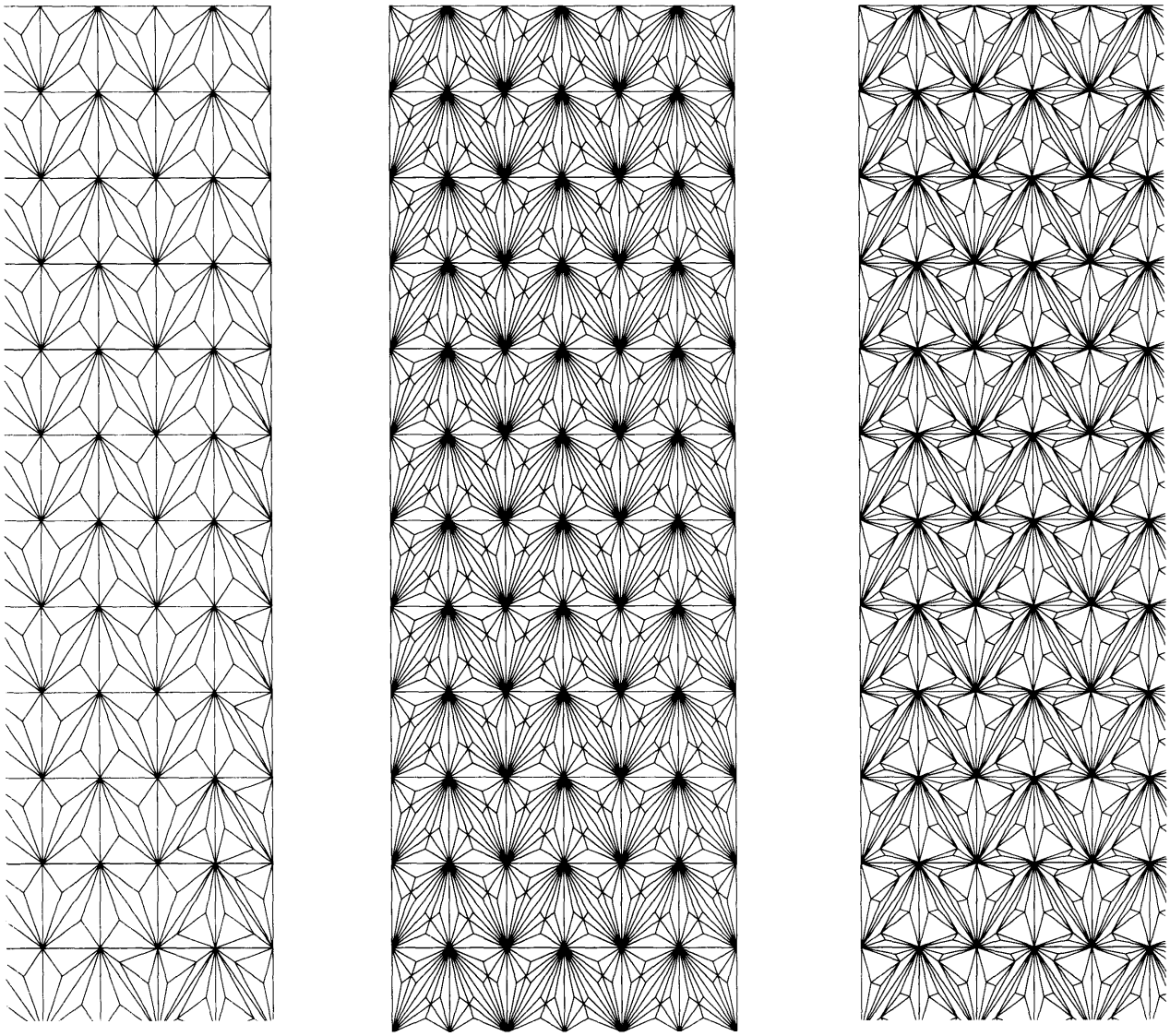


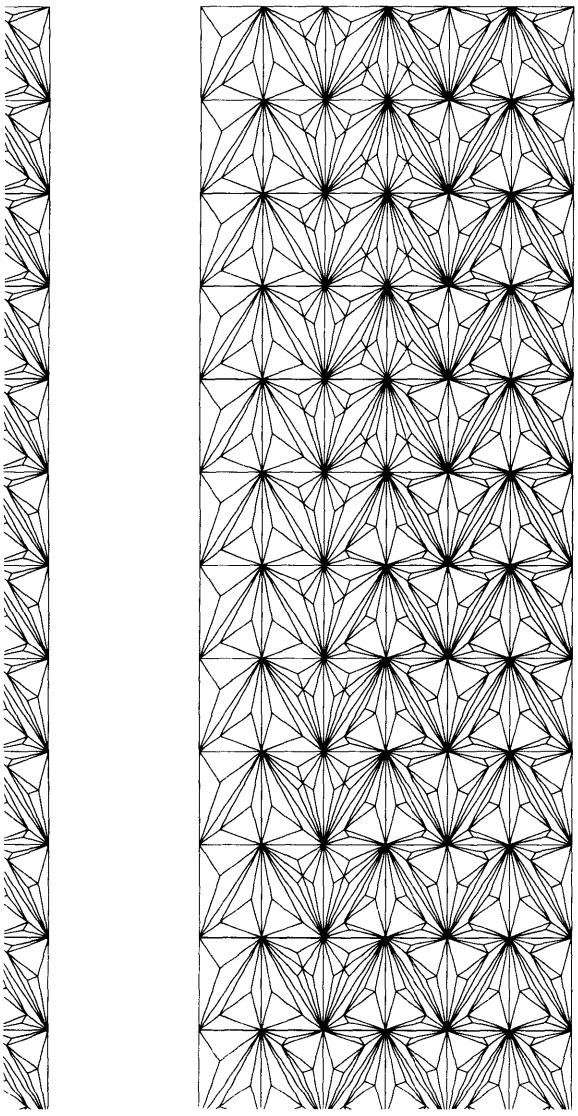




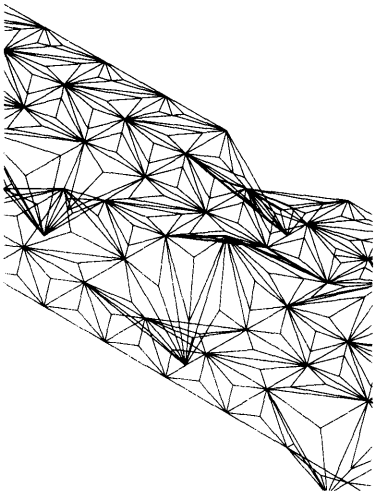
This feedback, however, is fairly limited in this study. The parameters are finite and a few runs yields a considerable sense of what the outcome might be. The pattern, it seems, does not have enough rules. The change in the pattern's form is described almost directly by the designer: 4 columns, 5 rows, 8 units apart.

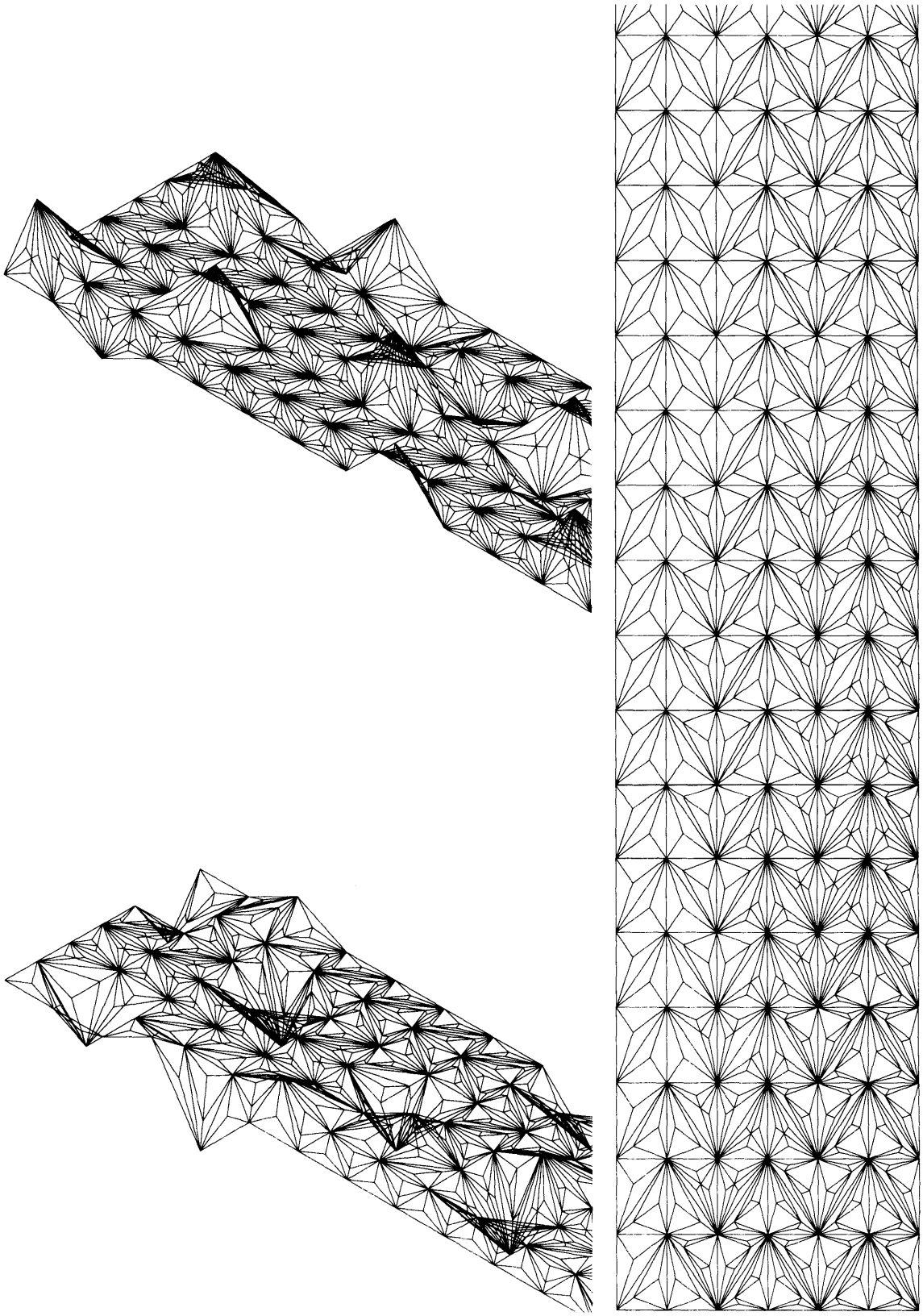


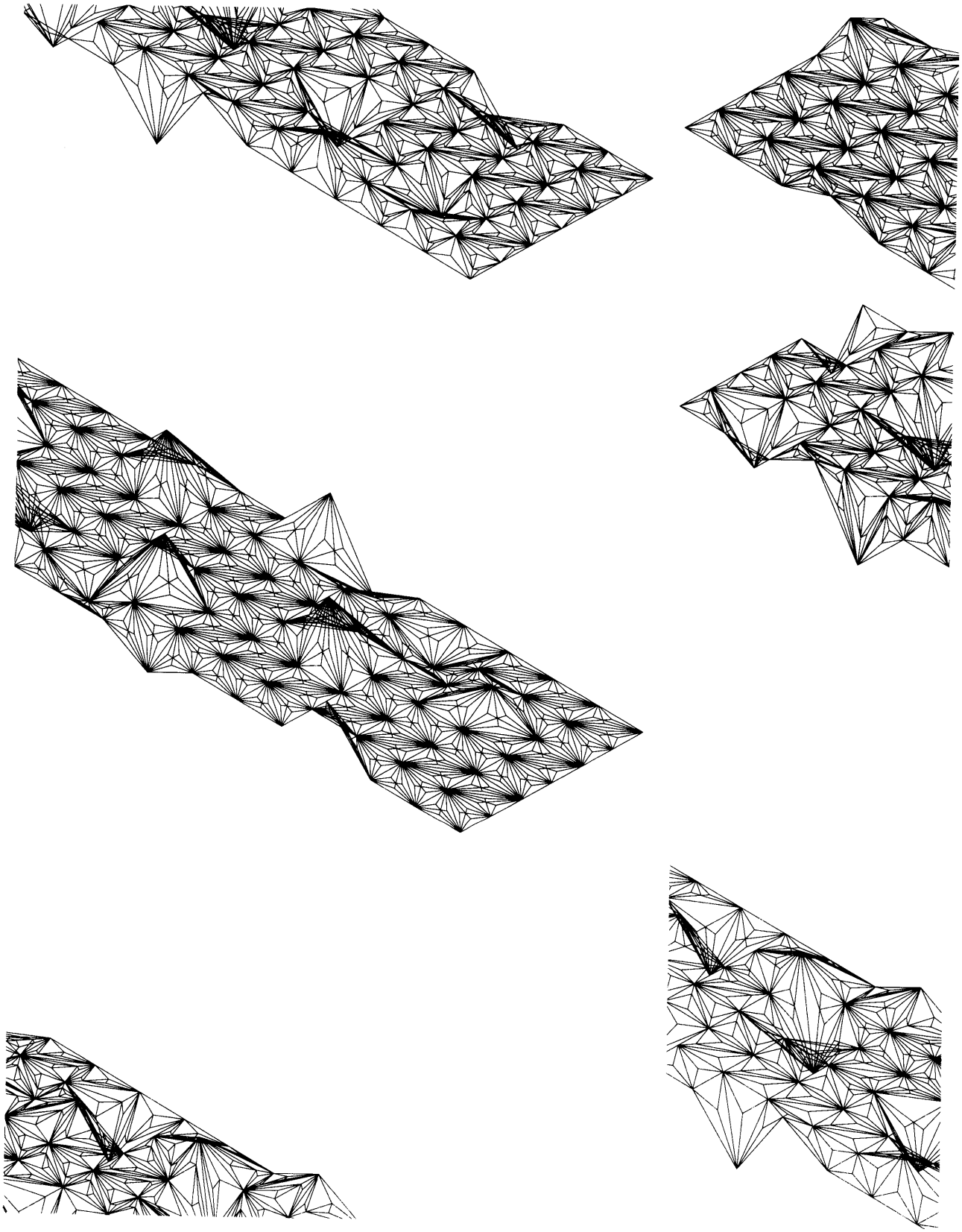


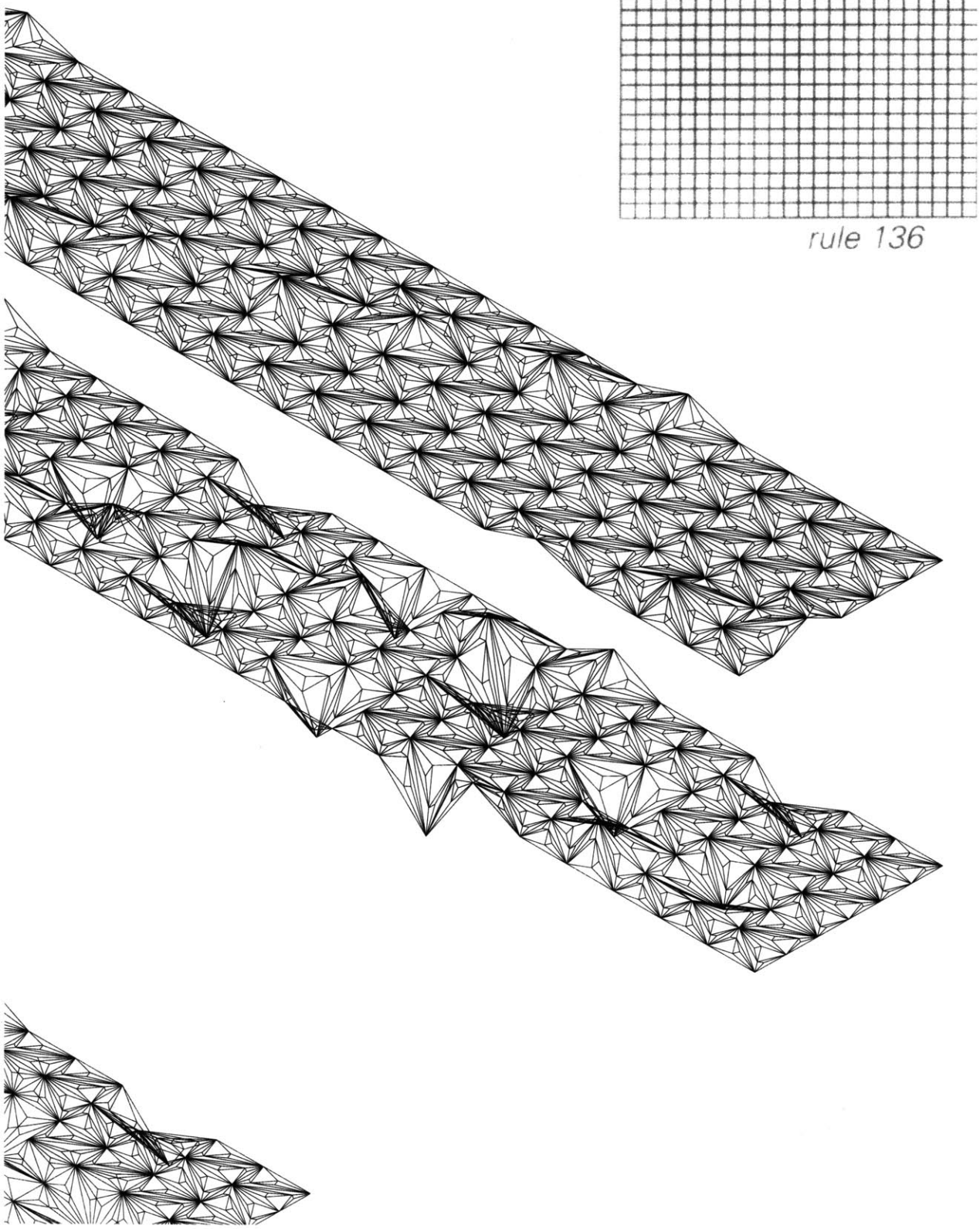


The counter-rhythm established in the three-dimensional plane is also designer-established: raise every 11^{th} point, every 5^{th}. It would be possible to get different numbers running against each other and more work could be done with this. However, in order to pursue this relationship it became necessary to suspend unit descriptions and focus on pattern definition alone.

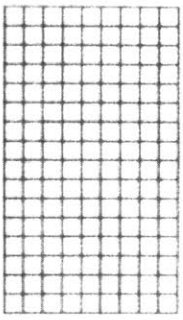


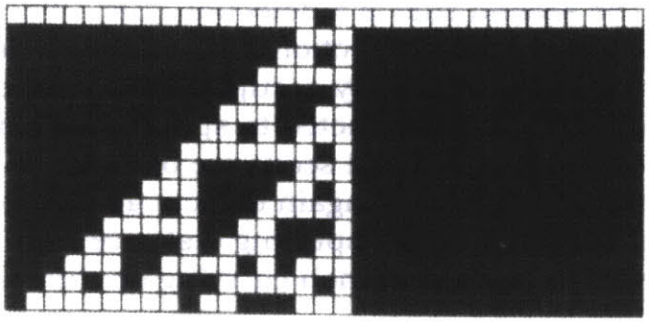





rule 136

6 *Point-Line*



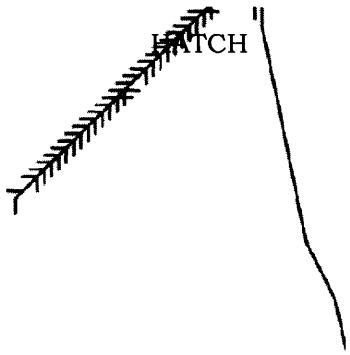


rule 137

Recent studies of cellular automata were used to understand reciprocally generating forms, forms defined by competing rules playing out against one another. Taking this step releases the designer from all agency except the seed-definition: the definition of the first cell and what it is to do.

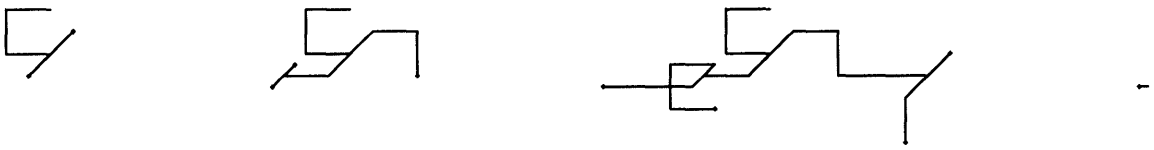
The final forms seen here have no manual adjustments made to them as in *Hexagon, Hatch* and *Diamond*. Nor are they derived from singular descriptions of a fixed set of final forms as in *Triangle*. Instead they are generated through the execution of a few simple rules, predefined rules which had no specific shape in mind when they were defined. Rather, the rules expressed an idea about spatial relationships.

First, the point "draws" a line in one of four directions. The choice of direction is determined through a counting sequence cycling through various combinations. For example, the counting sequence might start with "2". "2" draws in one direction. The program is told to go from "2" to "7". "7" draws in a different direction. "7" then directs the count to "8". Each direction has two numbers assigned to it. This further offsets the count. The goal of this is to have the lines drawn in a counter-intuitive sequence.



The second rule is a detection rule. When instructed to draw in one direction if the program detects that this point is already occupied by a start or end point of a line it is instructed to split in two points, and thus, an additional sequence of lines. If, when splitting, the program again detects an occupied point, it then halves the splitting distance until finding a vacant point and then carrying on from there.

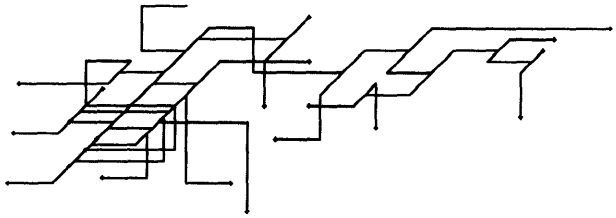
The strong 45 degree angles emerging in the pattern are a result of the splitting direction. When the points split, or double, they are told to move apart at this angle.



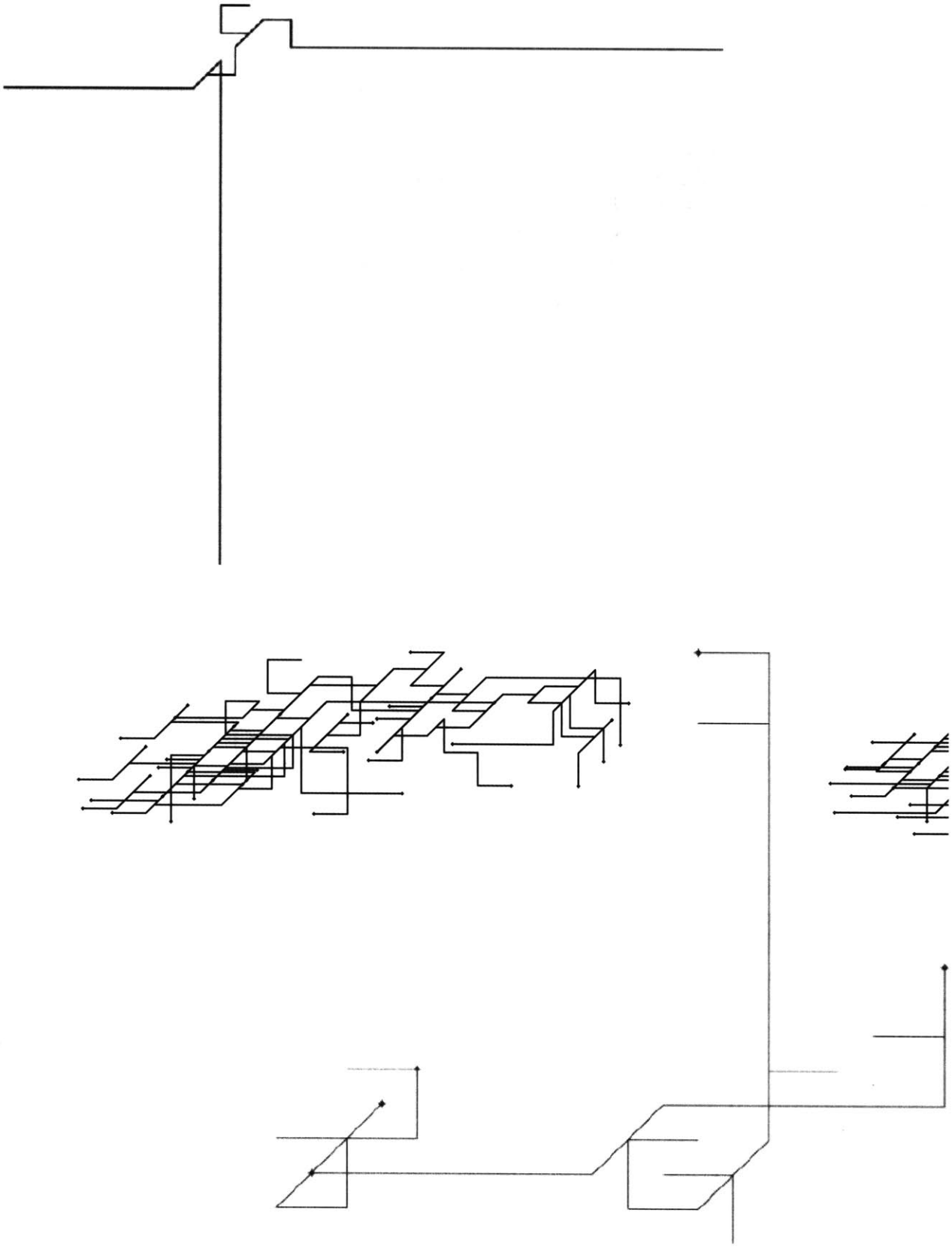
The spatial idea here is one of unique occupation of a location. It says, in a sense, that there is only room for one object per unit of space. However, it encourages density. The splitting distance is half the travel distance. Therefore, although each point may have its own "land" it has a number of close neighbors.

It is also represents an idea about obstacle relationships. It gives priority to previously established objects, moving around them as opposed to moving them.

The sequence of points forms a list, or family, of points. Each new point is admitted to the total group of points and each point in the group has the move-copy operation repeatedly performed on it. This has several effects. First, because of the counting sequence the motion of each point never follows the same pattern. As the group of points expands the counting sequence becomes repeatedly offset and staggered. This, then, gives some points the opportunity to "strike out" on their own: dispersing from the group, colliding with its path, and starting a new group.



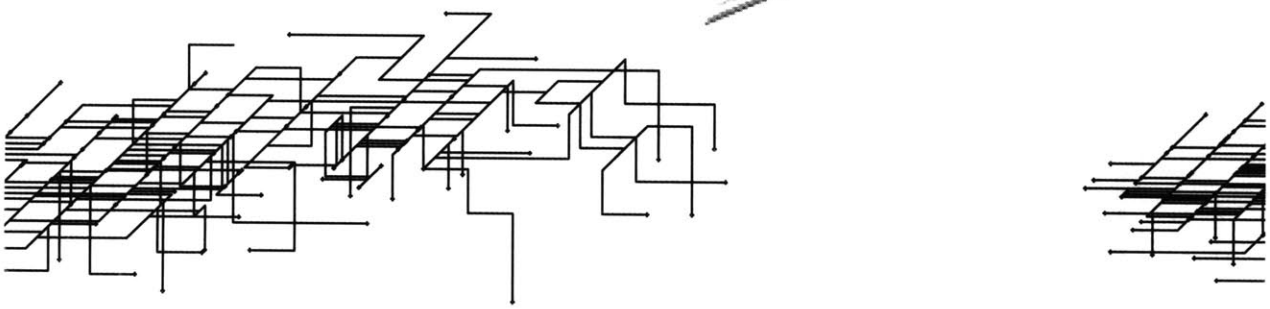
There are additional ideas to be drawn from this study: ideas about the nature of duplicated growth and density, ideas about random motion and growth. It is a development of a simple, traditional conception of Cartesian coordinate space: space governed by orthogonal unit increments.



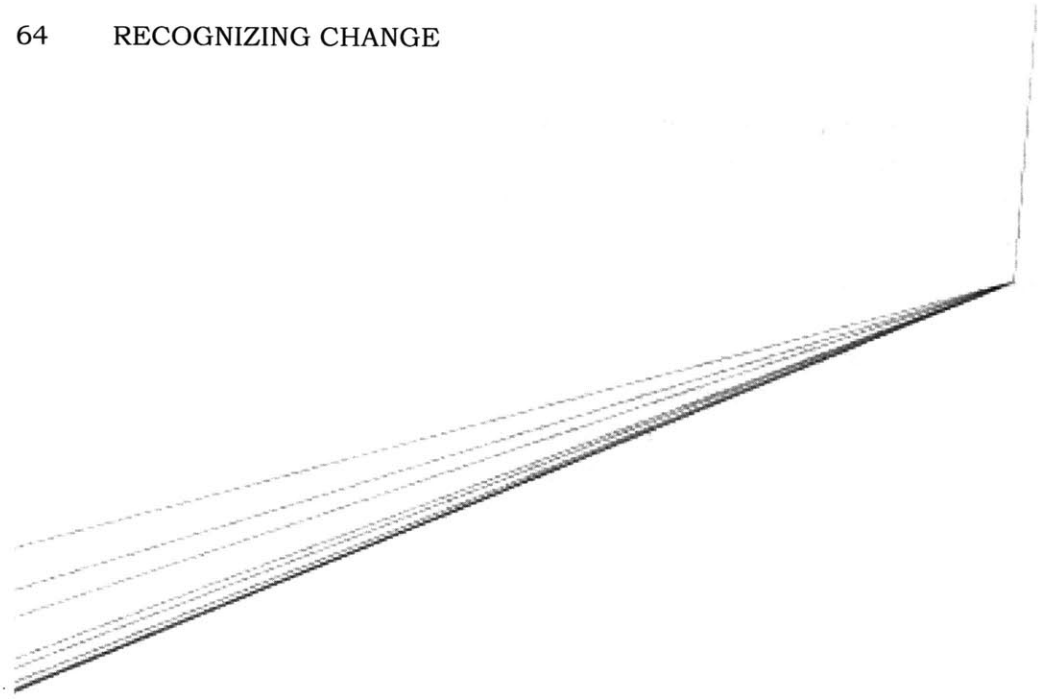
The pattern can be run successively and has a low probability of repeating the same form. However, because each rule is so constrained the final forms are clearly members of a "family" or type, a type sharing the same rules, but generating iterative forms.

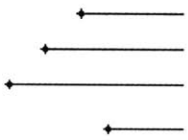
Generating rules at the root of the pattern definition suggests a different way to define pattern. Here, literal repetition happens only through the text of the script, or program. It is not directly seen in the pattern. But through the iterative observation of behavior – not singular, repeated form – a pattern is able to be recognized.

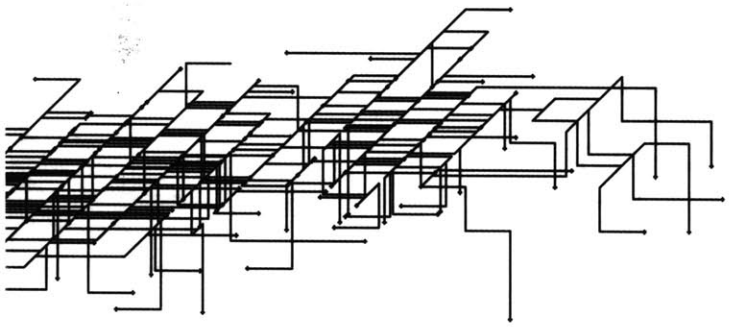
Pattern in this sense cannot be described through a single event, but is distinguished through repeated behaviors. Pattern here is a different kind of combination of perception and concept. The concept is

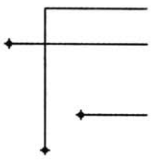


not seen through representation as in the incremental cell growth of *Hexagon* or even more traditional modes like the half-palmette motif. Rather, the concept is abstracted. It is self-consciously and self-referentially embedded in the rules of pattern-making itself. It simulates self-awareness.



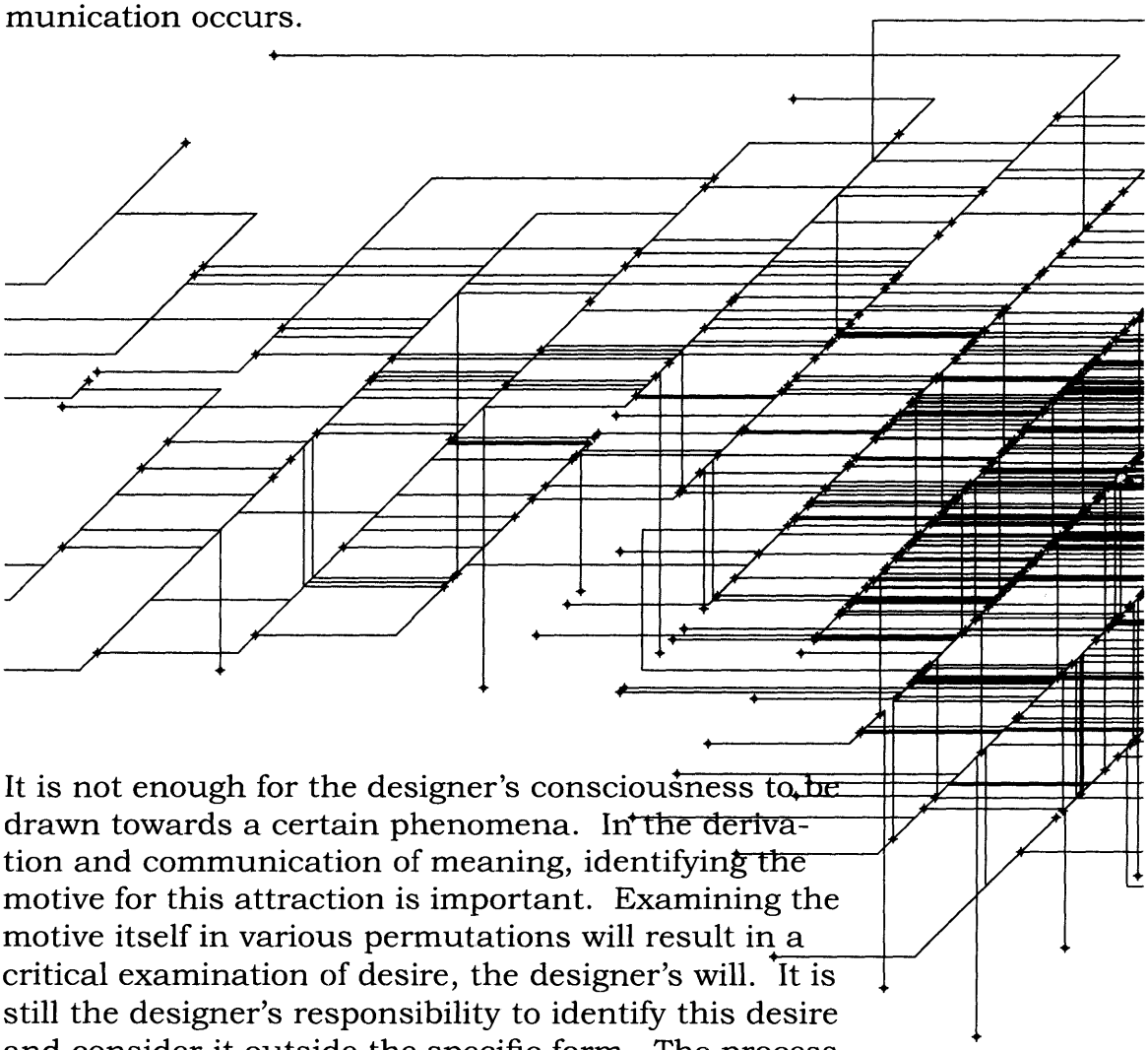




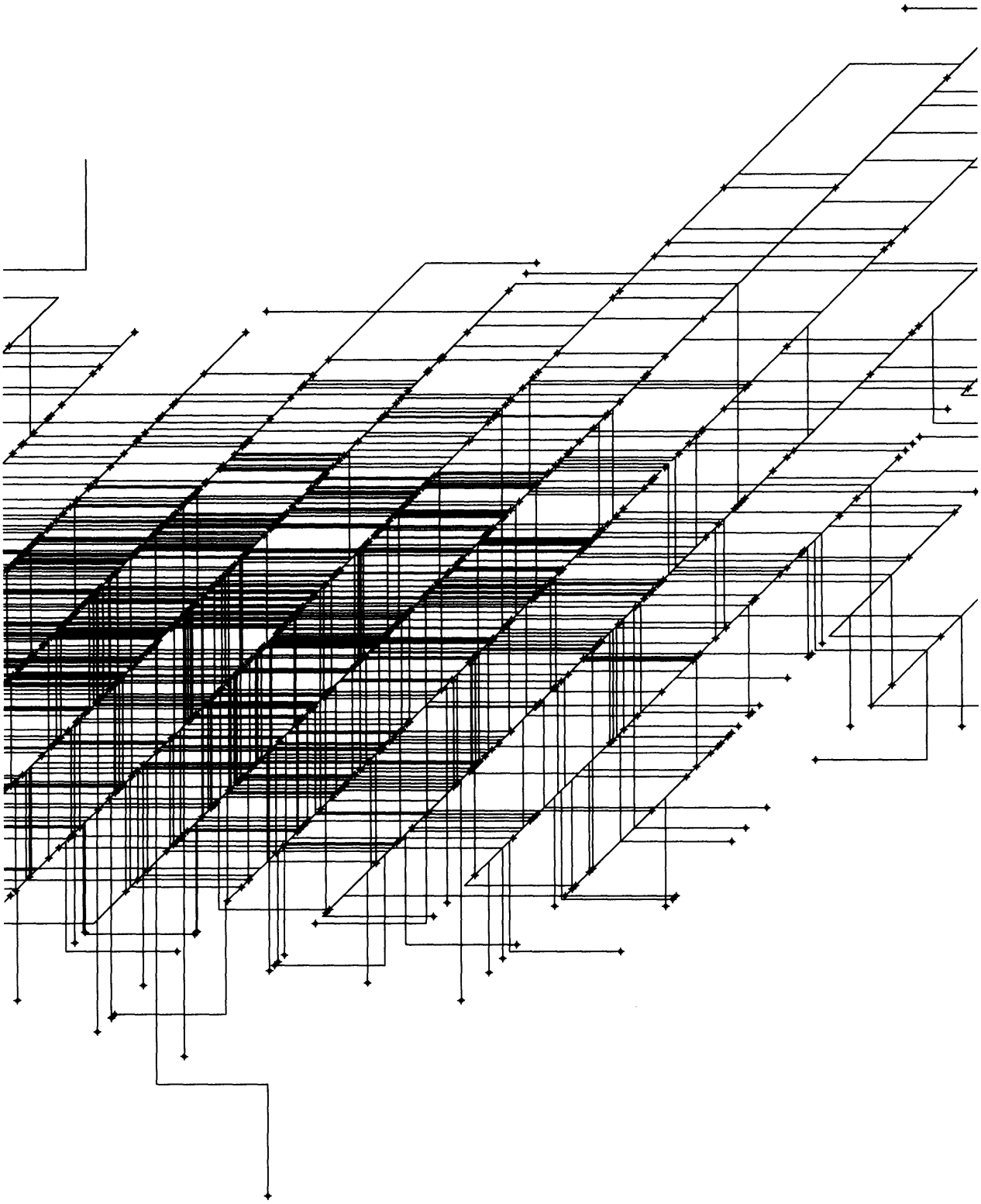


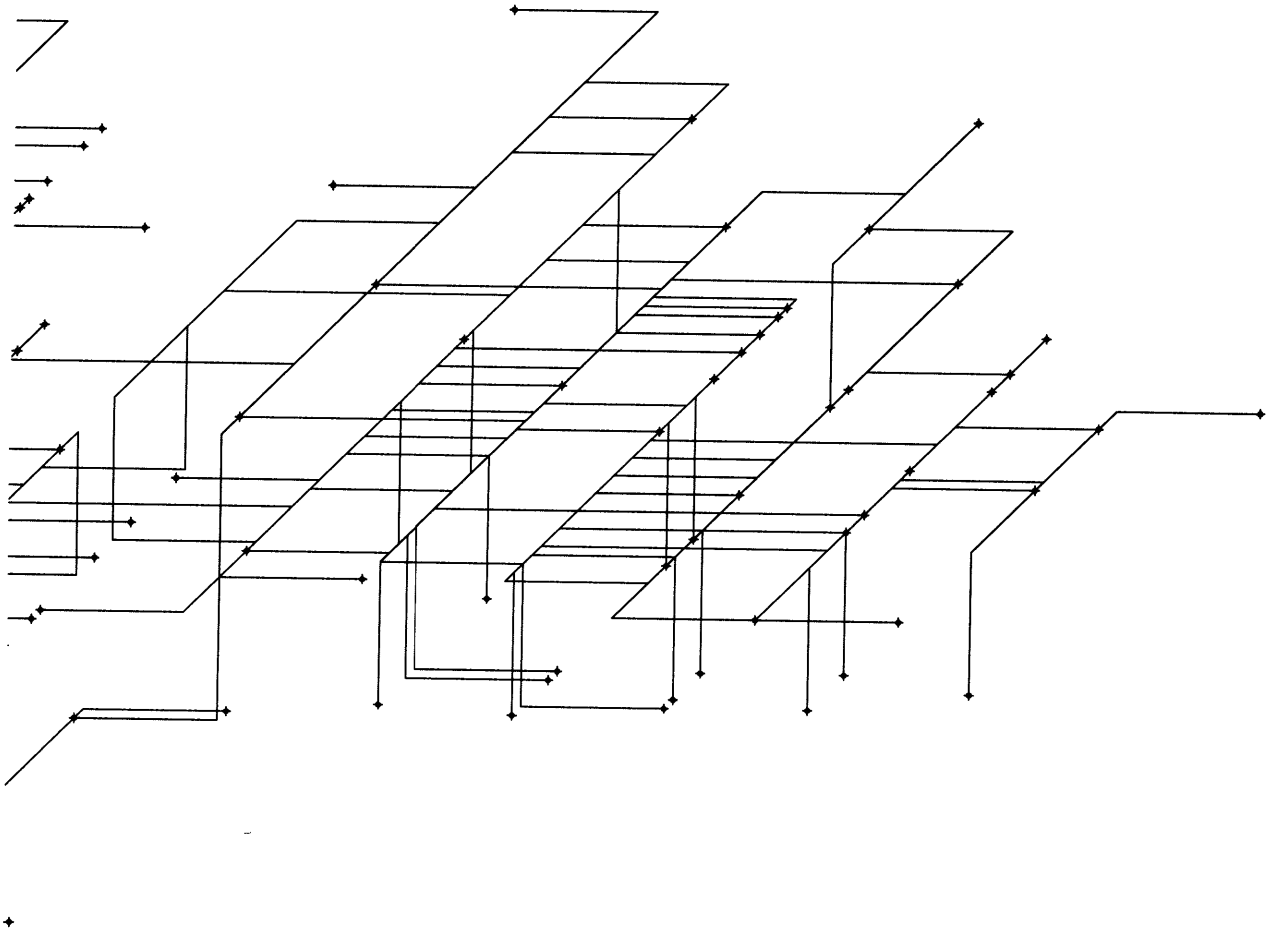
This exercise is a simple and fundamental demonstration of rule-based form. The rules here occur only at the level of pattern. They are not a part of the unit definition. The unit is an inherited element: a point and a line. Therefore there are two opportunities available here. One is to examine and develop unit-based rules, as well as to embellish and extend the "sensitivity" of the pattern.

But the primary opportunity is aesthetic: the link between the meaning of the rules and their perception. The move-copy sequence described here is arbitrary. Its meaning is largely formal. It is a vessel for visual description, not the result of careful consideration of the meaning latent in a certain rule sequence. It is in considering meaning, what the author chooses to formalize, what conditions their consciousness is drawn towards, that the job of communication occurs.



It is not enough for the designer's consciousness to be drawn towards a certain phenomena. In the derivation and communication of meaning, identifying the motive for this attraction is important. Examining the motive itself in various permutations will result in a critical examination of desire, the designer's will. It is still the designer's responsibility to identify this desire and consider it outside the specific form. The process of design grows from drawing external friction into the process, eschewing narcissistic fascination with the form alone.





Conclusion

This document is an essay on the rhetorical possibilities of pattern in today's culture. This work is an aesthetic gesture, a look at the connection between perception and meaning. It is a consideration of pattern's capacity to communicate and distribute meaning through tools and thoughts available to today's architectural culture.

Pattern has traditionally been a means for cultures to express and develop ideas about themselves. It provides a way for cultures to do this beyond the logic of written language, using instead the codes and processes of visual perception. The transcription and production of these codes rely heavily on the technology available to a culture. This necessity creates a continual feedback between culture and technology.

Recent developments in these social areas – the description of a psychological consciousness and the extraordinary expansion of computational capabilities – necessitate a response or speculation as to the particular nature and character of today's feedback relationship. This document represents one such speculation.

The primary characteristic of this relationship that this thesis explores is that of change. Change is a common component of any feedback relationship between technology and culture. However, the heightened sensitivity of psychological awareness and the accelerated responses of computational technologies distinguish today's cultural changes. Fluctuating change characterizes the loop between ideas introduced by psychology and the methods of assimilating those ideas suggested by computation.

Accelerated repetition and variation in today's culture also reduce the communicative value of singular, stable forms. Instead, communication and ideas are effectively expressed through iterative forms. The meaning of these forms is not latent in the perception of a singular object. Rather, meaning is taken from perceiving the rules and attributes that produce distinct objects within a larger genus or family of like, but individually unique, objects.

The production of pattern, this traditional means of cultural expression, retains and even gains considerable relevance within this kind of framework. Pattern is equally governed by attribute-transformation relationships. Throughout time these relationships have established the foundation for iterative designs: singularly unique designs growing from the same fundamental origins. This work seeks to intensify the iterative potential of pattern design, using external controls and computational methods.

Like the underlying grid of Moroccan Arabesques presented at the beginning of the chapter *Triangle*, or the families of rotated squares mentioned in the *Introduction*, a first visual gesture can create a family of perceptions that originate from a fundamental organization. Further, as demonstrated in the example of the half-palmette motif, the process of pattern distinction has a reciprocal effect on how first principles are described. The addition of the scroll to single half-palmette motifs was a gesture gradually absorbed as a first rule in the generation of all half-palmette design. In this way a trait that once distinguished an object from its cousins came to distinguish the larger family of designs itself (Schafter 47).

The distinctions and feedback suggested by the tradition of pattern design fold comfortably into the rapid succession of computational forms and the highly sensitive processes of distinction established by psychoanalysis. Psychology distinguishes classes

of conscious and subconscious conditions. These distinctions are made through behavior, through underlying principles or rules governing iterative and repeated actions. Not the form of a singularly distinct object or event.

The exercises presented here are not intended to be understood as a design tool. They are not intended to be replicated or didactically instructive. This is not a design manual. It is hoped that certain ideas – certain rules and attributes – may be freely combined and played out in the future by the author or others. But the exercises themselves stand out as iterations in a larger project of pattern research.

This document evidences certain efforts to guess at what such a confluence of computational technology and psychological awareness might lead to through the vehicle of pattern. The independent exercises are designs. These designs, in turn, reveal rules about pattern formation. These are rules of pattern generation that are useful in processing ideas about technology and culture. They are rules which can lead to processes of identity formation in a culture.

For instance, the rules of rotation and line assigned in *Hexagon* establish the forms as a family or class of designs. In *Hexagon* the gradual shift from dense to sparse cells is distinguished through the diagram of cell-division. This diagram derives from a specific angular relationship between lines: they must be, at any one instance, within 120° of one another (Thompson 592). Although this is not true for every angle in *Hexagon*'s patterns, the predominance of this relationship distinguishes the designs.

Such formal identity is also mutable, able to serve different spatial situations. A form's repeated, flexible use becomes an identifying or distinguishing trait of cultures that use it. Debra Schafter cites one such example in the origin and associations of the half-palmette:

"The artistic need to fill corners created by undulating scrolls gave rise to perhaps the most important modification – the half-palmette. More than any other single motif, the half-palmette represented to Riegl the enormous flexibility of a conventionalized motif as it responded to artistic invention. For him, it also provided the means by which to extend [a] ... conventionalized motif ... into [an] ... ornamental vocabulary of late antiquity" (Schafter 48).

This thesis proposes to extend certain conventions of pattern-making into an ornamental vocabulary of contemporary culture: a culture characterized by iterative behaviors and forms. The incremental changing of form, as in *Hexagon*, within a single pattern indicates an intertwining of rules and transformations. This intertwining makes possible mutable and contextual forms. Like the cells of the skin change shape and size in response to the skeletal construction of the elbow, incrementally changing surface units provide a way to account for change, as opposed to clashing with it.

Pattern's framework of attribute-transformation connects the psychological description of behavior and the computational description of form. Through pattern design becomes a framework for connecting ideas and perceptions by brining change into physical form. This connection -- the act of reaching into the water, the activity of design -- is more than a way to answer physical needs. In meeting these needs design becomes a way to work change over in our mind. It becomes not only the eyes of culture and the texture of technology. It enters the mind of a society struggling to negotiate the forces of change.

References

Castéra, Jean-Marc. *Arabesques: Art Décoratif au Maroc.* ACR Édition Internationale, Courbevoie (Paris), 1996.

Ching, Frank. *Design Drawing.* J. Wiley, New York, 1988.

Doshi, Dr. Saryu. *The Impulse to Adorn: Studies in Traditional Indian Archtiecture.* Marg Publications, Bombay, 1982.

Flagge, Ingeborg; Herzog-Loibl, Verena; Meseure, Anna. *Thomas Herzog: Architecture and Technology.* Prestel, New York, 2001.

Gombrich, E. H. *The Sense of Order: A Study in the Psychology of Decorative Art.* Phaidon Press Limited, Oxford, 1984.

Janson, H.W. and Anthony F. *History of Art. Sixth Ed.* Henry N. Abrams, Inc., New York, NY, 2001.

Levine, Peter A. *Waking the Tiger: Healing Trauma.* North Atlantic Books, Berkeley, CA, 1997.

McLuhan, Marshall. *Understanding Media: The Extensions of Man. Critical Ed.* Gingko Press, Corte Madera, CA, 2003.

Ovid. *The Metamorphoses of Ovid.* Johns Hopkins University Press, 1994.

Paccard, André. *Traditional Islamic Craft in Moroccan Architecture.* Éditions Atelier, Saint-Jorioz, France, 1980.

Pevsner, Nikolaus. *Ruskin and Viollet-le-Duc: Englishness and Frenchness in the Appreciation of Gothic Architecture.* Thames and Hudson, London, 1969.

Ruskin, John. *Modern Painters. vol. 1-5.* Smith, Elder and Co., London, 1873.

Sandler, Joseph; Person, Ethel; and Fonagy, Peter. *Freud's "On Narcissism: An Introduction" (with facsimile of "On Narcissism").* Yale University Press, New Haven and London, 1991.

Schafter, Debra. *The Order of Ornament, the Structure of Style: Theoretical Foundations of Modern Art and Architecture.* Cambridge University Press, Cambridge, United Kingdom, 2003.

Silverstone, Roger. *Proper Distance: Towards and Ethics for Cyberspace.* Essay, 2004.

Steinfeld, Kyle. *Will You Have a Yam?: A Study in Agency and Representation.* MIT, Cambridge, MA, 2004.

Thompson, D'Arcy Wentworth. *On Growth and Form. The Complete Revised Ed.* Dover Publications, Inc., New York, 1992.

Tichenor, James. *Electronically Modulated Materials: Effects and Context.* MIT, Cambridge, MA, 2004.

Venning, Barry. *Turner.* Phaidon Press Limited, London, 2003.

Wolfram, Stephen. *A New Kind of Science.* Wolfram Media, Inc., Champaign, IL, 2002

Image Reference

Introduction

Castéra p.311.

Schafter p.46. Alois Riegl, profile view of palmette, from Riegl, Stilfragen.

Schafter p.46. Alois Riegl, combined view of palmette, from Riegl, Stilfragen.

Schafter p.49. Palmette under-handle motif on Apulian red-figure vase, from Riegl, Stilfragen.

Schafter p.51. Example of Arabesques in manuscript illumination from Cairo, 1411, from Riegl, Stilfragen.

Castéra p.138.

Schafter p.21. John Ruskin, drawing, *Part of the Cathedral of St. Lo, Normandy,* from Ruskin, *The Seven Lamps of Architecture,* plate II.

Pattern Studies

Hexagon

Thompson p.596. Six cells, three connecting walls arranged three different ways.

Thompson p.596. Various possible arrangements of intermediate positions, in groups of 3, 4, 5, 6, 7 or 8 cells.

Hatch

Ching. Demonstration of cross-hatching.

Janson p.431. detail. Leonardo da Vinci. Adoration of the Magi. 1481-82.

Diamond

Flagge p.168. Laser demonstration of daylight grid system. Design, Thomas Herzog.

Flagge p.168. Section detail of daylight grid system.

Triangle

Paccard p.209. Tracing over tile motif.

Doshi p.131. The systematic breakdown of the fundamental region of hexagonal symmetry.

Paccard p.182. Tracing for ancient motif called "qarqba" (buskin). From zalijs at Marrakesh.

Point-Line

Wolfram p.54. Rules 136 and 137 from evolution of cellular automata with a sequence of different possible rules, starting in all cases from a single black cell.

Appendices

```
(defun Program_4_varheight (num_cols
num_rows row_inc col_inc List2_inc height2
List3_inc height3)

        (command "ucs" "")
        (command "osnap" "off")
        ;write points
        (load"d:/stuff/academicmit/thesis/scripts/
program4_2count/load/P4_write_points")
        (setq Point_List1 (p4_write_points
num_rows num_cols row_inc col_inc))
        (command "zoom" "extents")
        ;setq list2
        (setq Point_List2 (list (nth 0 Point_List1)))
        (setq prime_counter1 1)
        (repeat (- (fix (/ (length Point_List1)
List2_inc)) 1)
                (setq Point_List2 (cons (nth
prime_counter1 Point_List1) Point_list2))
                (setq prime_counter1 (+ List2_inc
prime_counter1))
        )
        (setq prime_counter1 1)
        (print Point_List2)
        (setq Point_List3 (list (nth 0 Point_List1)))
        (repeat (- (fix (/ (length Point_List1)
List3_inc)) 1)
                (setq Point_List3 (cons (nth
prime_counter1 Point_List1) Point_list3))
                (setq prime_counter1 (+ List3_inc
prime_counter1))
        )
        ;move setq
        (setq temp_ent_counter 0)
        (repeat (- (length Point_List2) 1)
                (setq temp_ent (nth
temp_ent_counter Point_List2))
                (command "move" temp_ent "" (list 0
```

```
0 height2) "")
                (setq temp_ent_counter (+ 1
temp_ent_counter))
        )
        (setq temp_ent_counter 0)
        (repeat (- (length Point_List3) 1)
                (setq temp_ent (nth temp_ent_counter
Point_List3))
                (command "move" temp_ent "" (list 0 0
height3) "")
                (setq temp_ent_counter (+ 1
temp_ent_counter))
        )
        (print "move done")

        ;draw cells (other conditional based on angle?)

        (load "d:/stuff/academicmit/thesis/scripts/
program4_2count/load/AB_BArows")

        (AB_BA_rows Point_List1 num_rows num_cols
col_inc)
)
```

```
(load "d:/stuff/academicmit/thesis/scripts/
program_point-line/p13_goodrun")

;;;;;;;;;;;;;;;;;;;;;;;
(setq counter_1 0)

(setq point_coords '(0 0))
(setq mov_ptlist (make_point point_coords))
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
(setq pt_num2 0)
(repeat (length mov_ptlist)
(setq mov_pt (nth pt_num2 mov_ptlist))
                                    (setq data_list
(make_collision_detect_list mov_pt))
                                    (setq
mov_pt_coords (get_data_for_group_code 10
mov_pt))
                                    (setq x_coord (car
mov_pt_coords))
                                    (setq y_coord
(cadr mov_pt_coords))

(test_mov_to_coords data_list x_coord y_coord)

                                    (if (= copy_flag1
nil)

(move_pt_draw_line mov_pt test_x test_y x_coord
y_coord)
                                        (progn
                                            (setq
mov_ptlist

(copy_move_draw_lines mov_pt test_x test_y
mov_ptlist))
                                        )
                                    )
                                    (setq pt_num2 (+
```

```
1 pt_num2))
)

(command "zoom" "extents")
```

```
(load "d:/stuff/academicmit/thesis/
scripts/masterutilities/get_data_for_group_code")


(defun make_point (point_coords)
        (command "point" point_coords)
        (setq mov_pt (entlast))
        (setq mov_ptlist (list mov_pt))
)




(defun move_or_copy_pt_write_ln (mov_ptlist /
new_mov_ptlist)
        (setq pt_num 0)
        (repeat (length mov_ptlist)
                (setq mov_pt (nth pt_num mov_ptlist))
                (setq data_list (make_collision_detect_list
mov_pt))
                (setq mov_pt_coords
(get_data_for_group_code 10 mov_pt))
                (setq x_coord (car mov_pt_coords))
                (setq y_coord (cadr mov_pt_coords))
                (test_mov_to_coords data_list x_coord
y_coord)

                (if (= copy_flag1 nil)
                        (move_pt_draw_line mov_pt test_x
test_y)
                        (progn
                                (setq mov_ptlist
                                        (copy_move_draw_lines
mov_pt test_x test_y mov_ptlist))
                        )
                )
                (setq pt_num (+ 1 pt_num))
        )
        (print mov_ptlist)
)
```

```
(defun make_collision_detect_list (mov_pt / data_list)

        (setq pt_coords (get_data_for_group_code 10
mov_pt))
        (setq x_coord (car pt_coords))
        (setq y_coord (cadr pt_coords))
        (command "zoom" "c" (list x_coord y_coord) 3.0)
        (setq ss_detect
                (ssget "_C" (list (+ x_coord 1.01) (- y_coord
1.01)) (list (- x_coord 1.01) (+ y_coord 1.01)))
        )

        (setq ent_check (ssname ss_detect 0))
        (setq data_list10 (list (get_data_for_group_code
10 ent_check)))
        (setq ent_num 1)
        (repeat (- (sslength ss_detect) 1)
                (setq ent_check (ssname ss_detect
ent_num))
                (setq data_list10 (cons
(get_data_for_group_code 10 ent_check) data_list10))
                (setq ent_num (+ 1 ent_num))
        )

        (setq ent_check (ssname ss_detect 0))
        (setq ent_type (get_data_for_group_code 0
ent_check))
        (if (= "LINE" ent_type)
                (setq data_list11 (list
(get_data_for_group_code 11 ent_check)))
        )
        (setq ent_num 1)
        (repeat (- (sslength ss_detect) 1)
                (setq ent_check (ssname ss_detect
ent_num))
                (setq ent_type (get_data_for_group_code 0
ent_check))
                (if (= "LINE" ent_type)
                        (setq data_list11 (cons
(get_data_for_group_code 11 ent_check) data_list11))
                )
```

```
        (setq ent_num (+ 1 ent_num))
        )

        (setq data_list (append data_list10 data_list11))
        (print data_list)
)


(defun test_mov_to_coords (data_list x_coord y_coord
/ return_expression)

        (setq line_counter_read (open
"d:/stuff/academicmit/thesis/scripts/masterutilities/
program10_line_counter.txt" "r"))
        (setq read_string (read-line line_counter_read))
        (close line_counter_read)
        (setq string_var (substr read_string (strlen
read_string) 1))
        (setq line_count_num (atoi string_var))
        (setq counter_1 (+ 1 counter_1))
        (if (= 0 (rem counter_1 5))
                (setq line_count_num 6)
        )

        (if (or (= line_count_num 5) (= line_count_num
        3))
                (progn
                (setq line_counter_write (open "d:/stuff/
academicmit/thesis/scripts/masterutilities/
program10_line_counter.txt" "w"))
                (if (= line_count_num 5)
                        (write-line (strcat read_string "2")
line_counter_write)
                        (write-line (strcat read_string "4")
line_counter_write)
                )
                (close line_counter_write)

                (change_coords1 x_coord y_coord)
                (setq copy_flag1
                        (cond ((point_check data_list test_x
test_y) (change_coords2 x_coord y_coord)))
```

```
                )
                ;(setq copy_flag2
                ;       ;(cond ((point_check data_list test_x
test_y) (change_coords3 x_coord y_coord)))
                ;)
                ;(setq copy_flag3
                        ;(cond ((point_check data_list test_x
test_y) (change_coords4 x_coord y_coord)))
                ;)

                )
        )

        (if (or (= line_count_num 6) (= line_count_num
        1))
                (progn
                (setq line_counter_write (open "d:/stuff/
academicmit/thesis/scripts/masterutilities/
program10_line_counter.txt" "w"))
                (if (= line_count_num 6)
                        (write-line (strcat read_string "4")
line_counter_write)
                        (write-line (strcat read_string "5")
line_counter_write)
                )
                (close line_counter_write)

                (change_coords4 x_coord y_coord)
                (setq copy_flag1
                        (cond ((point_check data_list test_x
test_y) (change_coords1 x_coord y_coord)))
                )
                ;(setq copy_flag2
                        ;(cond ((point_check data_list test_x
test_y) (change_coords2 x_coord y_coord)))
                ;)
                ;(setq copy_flag3
                        ;(cond ((point_check data_list test_x
test_y) (change_coords3 x_coord y_coord)))
                ;)

                )
```

```
	)

	(if (or (= line_count_num 8) (= line_count_num
	2))
		(progn
		(setq line_counter_write (open "d:/stuff/
academicmit/thesis/scripts/masterutilities/
program10_line_counter.txt" "w"))
		(if (= line_count_num 8)
			(write-line (strcat read_string "1")
line_counter_write)
			(write-line (strcat read_string "3")
line_counter_write)
		)
		(close line_counter_write)

		(change_coords3 x_coord y_coord)
		(setq copy_flag1
			(cond ((point_check data_list test_x
test_y) (change_coords4 x_coord y_coord)))
		)
		;(setq copy_flag2
			;(cond ((point_check data_list test_x
test_y) (change_coords1 x_coord y_coord)))
		;)
		;(setq copy_flag3
			;(cond ((point_check data_list test_x
test_y) (change_coords2 x_coord y_coord)))
		;)

		)
	)

	(if (or (= line_count_num 7) (= line_count_num
	4))
		(progn
		(setq line_counter_write (open "d:/stuff/
academicmit/thesis/scripts/masterutilities/
program10_line_counter.txt" "w"))
		(if (= line_count_num 7)
			(write-line (strcat read_string "1")
line_counter_write)
```

```
                    (write-line (strcat read_string "8")
line_counter_write)
                )
                (close line_counter_write)

                (change_coords1 x_coord y_coord)
                (setq copy_flag1
                    (cond ((point_check data_list test_x
test_y) (change_coords2 x_coord y_coord)))
                )
                ;(setq copy_flag2
                    ;(cond ((point_check data_list test_x
test_y) (change_coords3 x_coord y_coord)))
                ;)
                ;(setq copy_flag3
                    ;(cond ((point_check data_list test_x
test_y) (change_coords4 x_coord y_coord)))
                ;)

                )
        )
)


(defun move_pt_draw_line (mov_pt test_x test_y
x_coord y_coord)
        (setq start_coords (get_data_for_group_code 10
mov_pt))
        (command "move" mov_pt "" (list (- test_x
x_coord) (- test_y y_coord)) "")
        (setq end_coords (get_data_for_group_code 10
mov_pt))
        (command "line" start_coords end_coords "")
)



(defun copy_move_draw_lines (mov_pt test_x test_y
mov_ptlist / mov_ptlist)
        (setq copy_dist 0.5)
        (setq start_coords (get_data_for_group_code 10
mov_pt))
```

```
        (setq x_coord (car start_coords))
        (setq y_coord (cadr start_coords))
        (setq copy_x (+ x_coord copy_dist))
        (setq copy_y (+ y_coord copy_dist))

        ;(setq copy_data_list (make_collision_detect_list
mov_pt))


        (while (= (point_check data_list copy_x copy_y)
T)
                (setq copy_dist (/ copy_dist 2))
                (setq copy_x (+ x_coord copy_dist))
                (setq copy_y (+ y_coord copy_dist))
        )




        (command "copy" mov_pt "" (list copy_dist
copy_dist) "")
        (setq mov_pt2 (entlast))
        (setq mov_ptlist (cons mov_pt2 mov_ptlist))
        (setq end_coords (get_data_for_group_code 10
mov_pt2))
        (command "line" start_coords end_coords "")
        (command "move" mov_pt "" (list (* -1 copy_dist)
(* -1 copy_dist)) "")
        (setq end_coords (get_data_for_group_code 10
mov_pt))
        (command "line" start_coords end_coords "")
        (print mov_ptlist)
)




(defun change_coords1 (x_coordt y_coordt)
                (setq test_x (- x_coordt 1))
                (setq test_y y_coordt)
                (print test_x)
                (print test_y)
```

```
)

(defun change_coords2 (x_coordt y_coordt)
        (setq test_x x_coordt)
        (setq test_y (+ y_coordt 1))
)

(defun change_coords3 (x_coord y_coord)
        (setq test_x x_coord)
        (setq test_y (- y_coord 1))
)

(defun change_coords4 (x_coord y_coord)
        (setq test_x (+ x_coord 1))
        (setq test_y y_coord)
)



(defun point_check (data_list test_x test_y / flag
data_list2)
    (setq flag nil)
    (setq data_list2 data_list)
    (while (and (null flag) data_list2)
        (setq point (car data_list2))
        (if (and
            (= test_x (car point))
            (= test_y (cadr point))
        )
        (setq flag T)
        (setq data_list2 (cdr data_list2))
        )
    flag)
)
```